Investigating Cursor-based Interactions to Support Non-Visual Exploration in the Real World

Anhong Guo[1,2], Saige McVea[1], Xu Wang[2], Patrick Clary[1], Ken Goldman[1], Yang Li[1], Yu Zhong[1], Jeffrey P. Bigham[2]

[1]Accessibility Engineering, Google Inc., Mountain View, CA, USA
[2]Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
{anhongg, xuwang, jbigham}@cs.cmu.edu, {saigem, pclary, kengoldman, liyang, yuzhong}@google.com

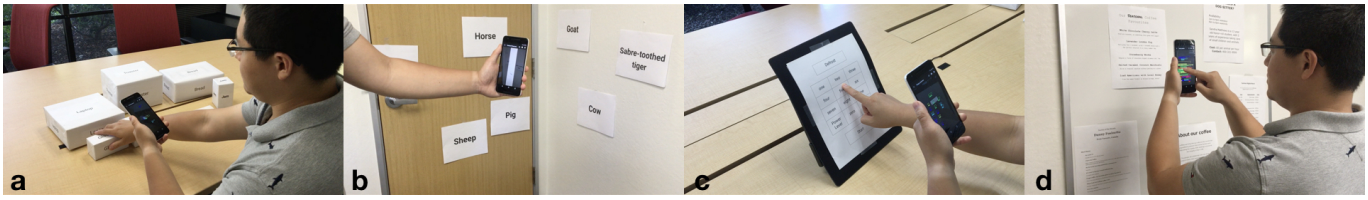

Figure 1. We define and compare cursor-based interactions that support non-visual attention to items within a complex visual scene: *(a)* window cursor, in which the user moves the device itself to scan the scene and receives information about what is in the center of the image; *(b)* vertical window cursor, a variation of window cursor that sacrifices the granularity on the vertical axis, but potentially facilitates locating the direction of a specific object; *(c)* finger cursor, in which the user moves their finger on the real world object they want to access and receives information about details near (or under) their fingertip; and *(d)* touch cursor, in which the visual scene is brought onto the device screen and the user moves their finger on the touchscreen to receive information about what they touch on the live camera image.

ABSTRACT

The human visual system processes complex scenes to focus attention on relevant items. However, blind people cannot visually skim for an area of interest. Instead, they use a combination of contextual information, knowledge of the spatial layout of their environment, and interactive scanning to find and attend to specific items. In this paper, we define and compare three cursor-based interactions to help blind people attend to items in a complex visual scene: window cursor (move their phone to scan), finger cursor (point their finger to read), and touch cursor (drag their finger on the touchscreen to explore). We conducted a user study with 12 participants to evaluate the three techniques on four tasks, and found that: window cursor worked well for locating objects on large surfaces, finger cursor worked well for accessing control panels, and touch cursor worked well for helping users understand spatial layouts. A combination of multiple techniques will likely be best for supporting a variety of everyday tasks for blind users.

Author Keywords

Non-visual exploration; cursor; interaction; accessibility; visually impaired; blind; computer vision; mobile devices.

ASSETS 2018, October 22–24, 2018, Galway, Ireland

ACM ISBN 978-1-4503-5650-3/18/10.
http://dx.doi.org/10.1145/3234695.3236339

INTRODUCTION

The development and prevalence of computer vision has brought tremendous changes to blind people's lives. For example, current computer vision systems can collect images taken by blind users as input, then analyze the images to produce an audio stream of information extracted (*e.g.*, Seeing AI, OrCam, etc.) However, visual scenes often contain large amounts of information. While an audio overview such as a scene description can be helpful as a summary, humans often need detailed information about specific parts of the visual scene. For blind people, focusing is not straightforward. Because they cannot see the image, they do not know what is contained within the image, and cannot simply visually skim and point to an area of interest. Instead, blind people use a combination of contextual information, knowledge of the spatial layout of their environment, as well as interactive scanning to find and attend to specific items [22]. For example, blind people apply this strategy for locating an object on the table, reading and accessing buttons on an appliance control panel, interpreting documents and signs, or learning the spatial layout of a scene.

Cursor-based interactions are defined by how users indicate a region of the image to attend to. This can be done in many ways, *e.g.,* by the current camera frame (or a region within it), by the location of a finger tip in the real world, or by a touch point on the device's touchscreen. Once a region is indicated, information and feedback relative to the cursor position are provided, *e.g.,* by speaking out the names of items or text in the cursor region. Users can explore based on the feedback. The cursor affects how easily they can query for certain types of information and within which types of visual scenes.

To explore this concept, we implemented three cursor-based interactions to help blind users attend to items within a complex visual scene (Figure 1), including: *(i)* window cursor, in which the user moves the device to scan the scene and receives information at the center of the camera, similar to VizWiz::LocateIt [3]; *(ii)* finger cursor, in which the user moves their finger on the real world object they want to access and receives information near their fingertip, similar to VizLens [8]; *(iii)* and touch cursor, in which the user moves their finger on the touchscreen and receives information of the relative location on the live camera image, similar to RegionSpeak [28].

Prior work has explored the concepts of the three cursor-based interactions individually, and suggested that different cursors might be more or less appropriate for certain tasks. In this paper, we contribute a study with 12 visually impaired participants where we first implemented the cursor-based interaction techniques, and formally compared the three techniques across a series of tasks that are representative of blind people's daily routines, including *(i)* locating an object in the environment, *(ii)* interpreting documents and signs, *(iii)* manipulating an appliance interface, and *(iv)* learning about their surroundings.

Our study results revealed that:

- Window cursor works well for locating objects on larger surfaces, but does not work well for small and fine-grained tasks. Blind users generally liked this one the most, as it was the simplest to use, and required the least amount of coordination.

- Finger cursor works well for accessing appliance control panels, but does not work well in pointing at remote objects in 3D space. Most users needed good instructions for this technique.

- Touch cursor works well for understanding the layout of a scene or document, but does not work well when mapping 2D screen locations is required to take actions on real-world objects (grabbing an object, pushing a button).

- A combination of multiple techniques will likely be best for supporting a variety of everyday tasks that blind users encounter.

The primary contributions of this paper are: *(i)* empirical results from a user study that expose the pros and cons of each technique on a variety of tasks based on real-world use cases, and *(ii)* design implications to apply and combine these techniques to support non-visual exploration. The study contributes understanding on how to best support blind people to extract visual information from the real world.

RELATED WORK

Our work is related to prior work on making visual information accessible with computer vision. The three cursor-based interactions that we define and study in this paper have been incorporated in various ways in prior research studies and products, although their use and trade-offs in different contexts have not been previously studied.

Computer Vision for Accessibility

The development and prevalence of computer vision has brought tremendous changes to blind people's lives. For example, current computer vision algorithms can collect images that blind users take as input, analyze the images, and produce an audio stream of extracted information as output.

Many systems have been developed to help blind people read visual text via OCR [6]. For instance, the KNFB Reader [12] is a popular application for iOS that helps users frame text in the camera's view, and then reads text that is captured. Other systems have been built to help blind people recognize faces [18, 19], identify products [14, 18, 19], count money notes [13, 18], or read the LCD panels on appliances [7, 16, 24].

Recently, deep learning approaches have been applied to general object recognition and scene description, in products such as Aipoly [1] and Microsoft's "Seeing AI" [19]. For example, Seeing AI [19] provides functionalities for blind users to take a picture and get an overview description of the captured scene.

While an overview such as a scene description can be helpful as a summary, humans often need focused information about one or more parts of the visual scene, which often contains large amounts of information. Generally, prior approaches have assumed that there would be a primary target in the camera's field of view. However, the interaction to attend to specific targets has not been made explicit. In response to this, prior work has explored various ways in assisting blind users attend to specific items within a complex visual scene.

Window Cursor Applications

A natural way to capture the information users want in a photograph is to move the camera around until the intended part of the photograph is contained within the frame. Several prior systems have been developed to help blind people take better photographs, since acquiring a high-quality photograph is often a prerequisite for further computer vision processing [9, 15, 25, 26, 27].

The challenge for these systems is both ensuring that some frame captured by the camera actually contains the object of interest, and developing an approach to alerting the user or automatically capturing the image when that occurs. For example, EasySnap [9, 26] reads out locations of faces and pre-registered objects in the field of view, and guides blind users to move their phone to take a better picture. VizWiz::LocateIt [3] allows blind people to ask for assistance in finding a specific object. Users first send an overview picture and a description of the item of interest to crowd workers, who outline the object in the overview picture. Computer vision on the phone then helps direct users to the specific object. In [25], an image composition model is used to provide aiming feedback, and the system automatically saves the best image captured. Scan Search [27] automatically extracts key frames from a continuous camera video stream, and identifies the most significant object inside the picture. This way, blind users can scan for objects of interest and hear potential results in real time.

Our window cursor interaction is similar to these techniques in that visually impaired users hold and move their phone to

scan the environment, then get real-time feedback about the objects and their locations in the field of view.

Finger Cursor Applications

Various projects have experimented with having visually impaired users use their fingers to access real world objects and information. In this interaction, the user's finger provides a direct connection to the item in the physical space.

Several projects use finger-worn cameras to read text and explore surroundings with computer vision. Fingerreader [21] assists blind users with reading printed text on the go with a finger-worn device. EyeRing similarly leverages a finger-worn camera to interpret immediate surroundings [17].

Other projects use cameras placed in the environment. Access Lens reads physical documents and lets a blind person listen to and interact with them [11]. In the direct touch interaction mode, Access Lens tracks the user's fingertip and speaks the text closest to it, which enables blind users to read previously inaccessible documents simply by touching them. Talkit [20] enables blind users to access 3D printed models with their finger and get audio cues about what is underneath their finger.

Using hand-held and head-mounted cameras, VizLens [8] fuses crowdsourcing and computer vision to interactively help blind people use inaccessible interfaces in the real world. Similar to a screen reader, VizLens provides feedback on what is beneath a user's finger. OrCam is a product that uses a head-mounted camera to make available various computer vision applications targeting low vision people [18]. Specifically, blind users point their finger at or on an object they want to recognize, then pull it away for a picture to be snapped.

The reason these projects used finger-based interactions is likely because the user's finger provides a direct connection to the item being explored and accessed in the physical space, and after locating the specific target, objects can be directly manipulated. Our finger cursor interaction is similar to these techniques, in that visually impaired users move their finger on the real world object they want to access, and get feedback about what is near their fingertip.

Touch Cursor Applications

Prior research has explored having visually impaired people use touchscreens to access mobile devices. In this interaction, the user drags a finger along the touchscreen or performs accessible gestures to navigate through information, or learn the spatial layout of documents and scenes.

Slide Rule developed multi-touch gestures that could control touchscreens non-visually [10], which informed the VoiceOver screen reader on iOS, and the TalkBack screen reader on Android. RegionSpeak [28] enables spatial exploration of the layout of objects in a photograph using a touchscreen. Users send a photo (or multiple stitched photos) to have the crowd label all of the objects in the photo. Users can then explore the photo on a touchscreen to learn about the spatial layout.

The reason these projects used touchscreen-based interactions is likely because the information can be easily represented digitally on the mobile device, fine-grained touch movements or accessible gestures can be used, and physically touching the object is not necessary or possible (due to proximity) for accessing the information. Our touch cursor interaction is similar to these techniques, in that visually impaired users drag one finger around the touchscreen to explore the content captured by the camera and mapped to the touchscreen dimensions.

To summarize, we implemented three cursor-based interaction techniques inspired by prior work for visually impaired users to get filtered and focused feedback from raw computer vision output. Different from prior work that focused on individual techniques to solve specific usage scenarios, we performed a thorough study to compare these techniques and understand their effectiveness for a variety of real-world tasks.

CURSOR-BASED INTERACTIONS

The cursor-based interaction concept involves: *(i)* indication of a cursor region, *(ii)* more focused information and feedback relative to the cursor, and *(iii)* further user exploration based on the feedback. We introduce three cursor-based interaction modes for different usage scenarios, created based on prior work. More specifically, a cursor is defined by three features: *(i)* cursor center location, *(ii)* cursor shape, which defines the region of the cursor, also drawn on the camera view, and *(iii)* cursor affinity function, which defines the relationship between the cursor/entity bounding boxes produced by computer vision models, and the conditions for entities to be read out.

Mobile Application

The mobile application is implemented in Java for the Android platform. The backend of the app runs a series of computer vision recognizers including object identification, face detection, landmark detection, food identification, optical character recognition (OCR), etc. These recognizers output entity bounding boxes with their labels, which are then read out sequentially through an audio stream. Since the speed of individual recognizers is quite slow (*e.g.*, OCR is ~1fps), optical trackers are used to maintain the rough locations of the bounding boxes between two processing frames.

We also included two earcons [4], which are brief and distinctive sound cues that provide additional context around the cursor region. When there are entities in the camera's field of view but are not worth being read out through Text-To-Speech (TTS), the "entity-in-view" earcon is played to indicate that the camera is generally aimed in the right direction. When the user's finger is in the field of view, the "finger-in-view" earcon is played to indicate that the camera and their finger are generally aimed in the right direction.

Window Cursor Mode

Window cursor mode announces entities in the center of the image. This potentially helps users find objects by moving their phone to scan the scene.

For this mode, the cursor center location is the center of the camera image, and the shape of the cursor is a small rectangle proportional to the camera image size. Entities are read out if the center of the entity is within the cursor bounding box. As illustrated in Figure 2a, the left entity is read out, while the right one is not. Note that the right entity also overlaps with

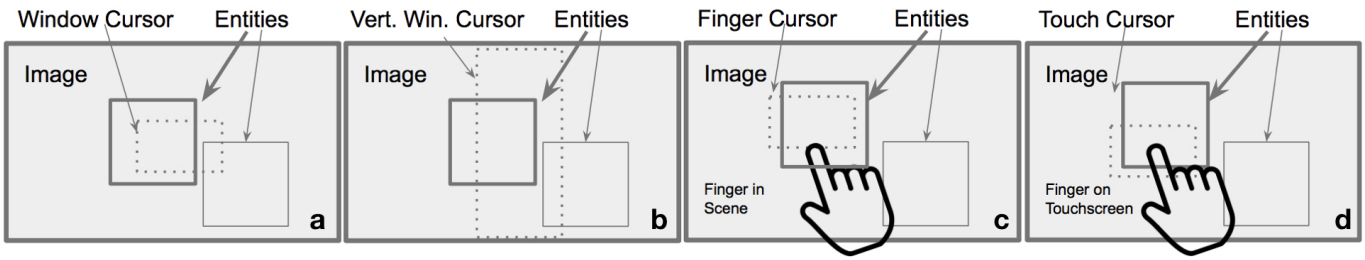

Figure 2. Illustrations of cursor-based interactions: *(a)* window cursor, *(b)* vertical window cursor, *(c)* finger cursor, and *(d)* touch cursor.

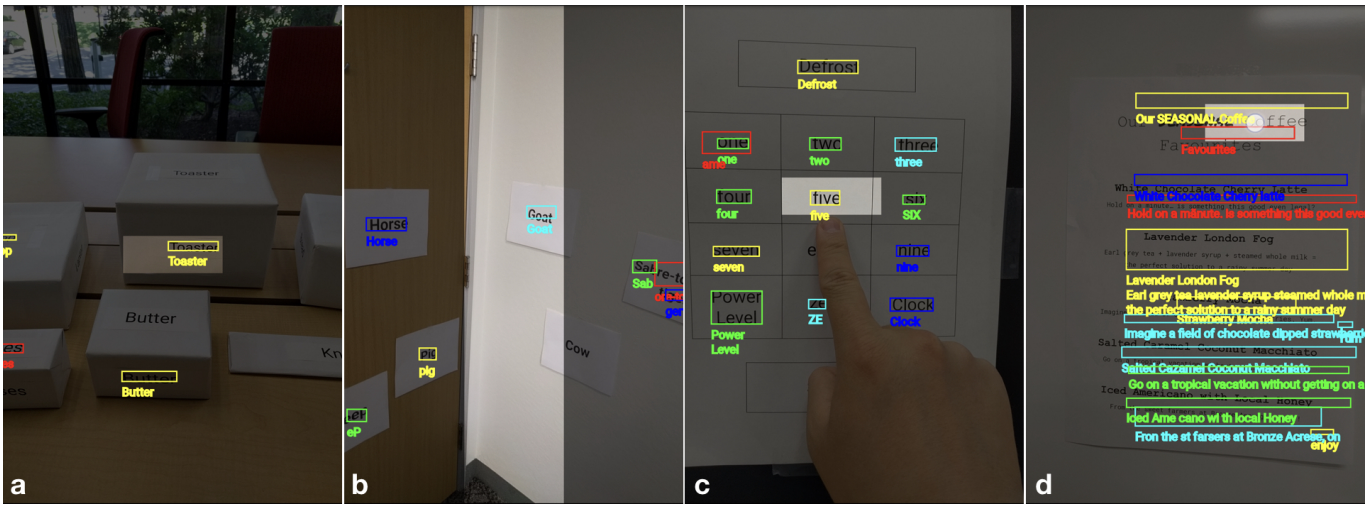

Figure 3. Screenshots of cursor-based interactions in different use cases: *(a)* window cursor, *(b)* vertical window cursor, *(c)* finger cursor, and *(d)* touch cursor. Colored bounding boxes show the recognized OCR results, the cursor region is drawn as a transparent overlay, while the rest of the image is covered with a semi-transparent dark overlay.

the cursor bounding box — if there is one entity bounding box that is much larger than the others and overlaps the others, then it is undesirable to always say the large one; while the center of focus should be the smaller ones. For example, if there are multiple objects (laptop, mouse, water bottle) on a table, when the user scans their device over the table, always speaking "table" may not be relevant to the task of locating the mouse. On the camera view (Figure 3), the cursor region is drawn as transparent, while the rest of the image is covered with a semi-transparent dark overlay.

Another variation of the window cursor shape is a vertical slit box in the middle of the screen (Figure 2b). This change sacrifices the granularity of entities on the vertical axis, but potentially makes it easier for the user to scan the scene in one direction to quickly locate the direction of a specific object the user wishes to find.

Finger Cursor Mode

Finger cursor mode announces entities near the user's fingertip in the scene. This potentially helps users identify objects, read documents, and use appliance control panels. This interaction is similar to OrCam's MyEye [18], and Access Lens [11].

For this mode, the center location of the finger cursor is above the user's topmost fingertip location in the camera image, and the shape of the cursor is a small rectangle proportional to the camera image size. The reason for the cursor location being above the fingertip is, when the user's finger is covering an object or a piece of text, our system will not be able to read it. For use cases such as appliance access, it might be more natural to provide direct feedback of what is underneath the finger, similar to how a screen reader works. Techniques such as keeping a memory of the entities in the scene and using reference images for computing homography would help to solve this problem (*e.g.*, VizLens [8]).

Entities are read out if the center of the entity is within the cursor bounding box. As illustrated in Figure 2c, the left entity is read out, while the right one is not.

Touch Cursor Mode

Touch cursor mode announces entities at the user's touch point on the screen. This potentially helps users explore and understand the spatial layout of a scene or document. The interaction is similar to the Explore by Touch in VoiceOver, TalkBack and RegionSpeak [28].

For this mode, the center location of the touch cursor is the user's touch point location on the touchscreen mapped to the camera image coordinate system, and the shape of the cursor is a small rectangle proportional to the camera image size. Entities are read out if the center of the entity is within the cursor bounding box. As illustrated in Figure 2d, the left entity is read out, while the right one is not.

USER STUDY

The goal of the user study was to better understand how the cursor-based interactions perform for various use cases. The user study sought to answer the following research questions:

- What are the strengths and limitations of each cursor?
- Which cursor is the most or least appropriate for each task?
- How well does the user build a mental model of the 3D visual scene from the auditory feedback?

Participants and Apparatus

We recruited 12 participants (6 male, 6 female) through online postings. Among the 12, 9 of them were blind and 3 were low vision users; 5 were in the age range of 25-34 and 7 were in the age range of 35-54; 11 had at least a bachelor's degree, and the other one had a professional diploma; 6 were currently working at a tech company, 4 were working at schools or banks, and 2 were not currently employed; 11 had used screen reader before; and 11 had been either blind or low vision for 18+ years, while the other one had been blind for 3-6 years.

We implemented the three cursor-based interaction modes and installed the application on an Android device. To control for recognizer performance in the study, we only included tasks that involved text labels, and only used an OCR recognizer to provide feedback to the user. We also kept the cursor regions of the techniques the same size for direct comparison. The users were provided with a plastic pouch to carry the device around their neck. However, they would decide whether or not to use it. We compared cursor methods within the same overall system, controlling for many variables, including cursor size, affinity function, and recognizer performance. Though they may not be optimal, our implementations were sufficient to generate useful in-depth insights about user experience.

Procedure

The user study contained three stages. In the first stage, we conducted discovery interviews to better understand each participant's background and needs. In the second stage, we asked each user to complete four tasks with the three cursor modes to observe the usability and pain points of each method. Following each task, we asked users about their experience completing the task and asked them to rate "Of the 3 methods for this task, which one did you most prefer? Least prefer?" In the third stage, after users had completed all tasks, we conducted semi-structured interviews to ask them about their overall experience. The interview sessions were audio-recorded and the task sessions were video-recorded.

Tasks

The task session took 35-45 minutes. We designed four tasks representative of daily activities people engage in using eyesight (Figure 4). Tasks were designed based on prior work [2, 8, 11, 15, 28], and in consultation with blind participants through pilot interviews, where we asked about daily tasks that were difficult to complete without sighted help, information they felt they were missing out on, as well as situations that made them feel curious about the environment. The scenarios were also confirmed by participants in study stage 1.

We then carefully designed the tasks to be authentic and involving edge cases. In the first task, participants were asked to locate a specific object on the table, such as glasses. We also intentionally included food and knife on a kitchen table so that participants would not use their hands to directly touch and explore as they normally do. In the second task, participants were asked to interpret documents and signs, *e.g.*, find the time and date of an event on a printed poster. For the third task, participants were asked to manipulate an interface, *e.g.*, press a button sequence on a flat and unlabeled appliance control panel. For the last task, participants were asked to learn about their surroundings, *e.g.*, explore an unfamiliar environment and identify what and where are the objects around them. These tasks generally require sighted assistance.

When blind people first come to an unfamiliar space, they need to learn about the surroundings. To effectively operate in a space, blind people need to first locate objects, and then interpret the objects or interact with them. Among the four tasks, environment exploration and object location requires 3D understanding and navigation, interface manipulation requires 2D understanding and navigation, and document interpretation requires the least spatial navigation.

We asked participants to complete the tasks with each of the three cursor methods (window cursor, finger cursor, and touch cursor). The sequence of the three cursor methods were counter-balanced across four tasks. All users completed the tasks in the same order. The task instructions are listed below.

Task 1: You are at a family gathering, and your aunt calls in after she left because she thinks that she forgot her glasses on the kitchen table. Could you find them for her?

Task 2: You've just stepped into a cafe for an iced drink, and the barista mentions they will be hosting live music this month to the person ahead of you. The schedule is on the bulletin board. Can you find and read the schedule?

Task 3: You've purchased a variety pack of fancy popcorn and each flavor needs to be warmed in the microwave for a very precise amount of time. The first requires 2 minutes and 49 seconds of cooking. Can you enter 2-4-9 on the microwave panel, then click the start button? Repeat for [3:17 + start], and again for [5:08 + start].

Task 4: A friend of yours is babysitting her niece for the day, and she invited you to join them at the petting zoo. You are standing in the centre of the park, and you hear her niece giggling. What animals are in the zoo? Where are they?

Methods

We took a qualitative approach when analyzing participants' responses in stage 2 and 3 of the study. Quantitative measures were used in prior work to evaluate individual cursors. We instead complement prior work with qualitative approaches, which is vital for gathering in-depth insights into user experience necessary for generating meaningful design implications.

We transcribed the video and audio recordings. We used a line-by-line coding approach [5] and synthesized the main concepts from the task sessions and subsequent interviews. We also

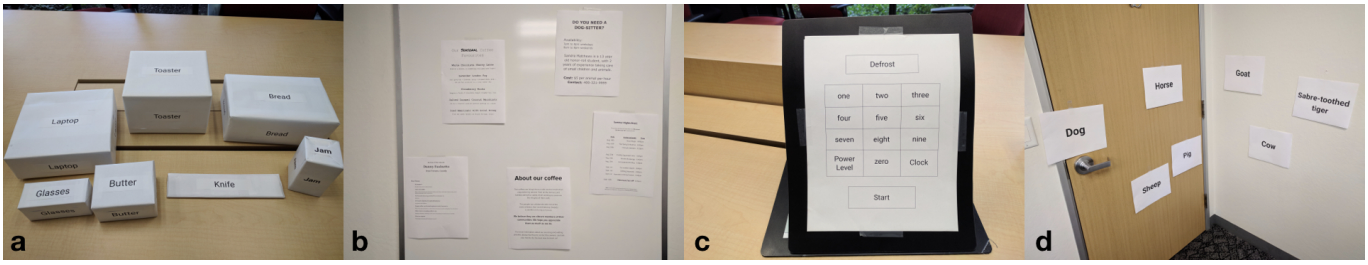

Figure 4. Study setup comparing the cursor techniques across a series of tasks representative of blind people's daily routines, including *(a)* locating an object in the environment, *(b)* interpreting documents and signs, *(c)* manipulating an appliance interface, and *(d)* learning about their surroundings.

selected user feedback quotes which are indicative of issues in the system and could inform future designs. Furthermore, we considered the diversity of participants (*i.e.*, not all quotes should come from P2), the diversity of problems addressed, and the clarity of meaning when selecting these quotes.

RESULTS

We now detail the user study results. We first present findings regarding the 9 blind users. For each task, we discuss user feedback and report preference rating responses for each cursor method. We then present results regarding the 3 low vision users. Finally, we summarize the key takeaways.

Task: Locate an Object

In this task, we aim to answer the question, "Can users leverage window cursor, touch cursor, or finger cursor to locate an object?" The results suggest that window cursor is the best among the three, with 7 most prefer, 1 neutral, and 1 least prefer in response to the preference rating.

Window Cursor

Users enjoyed that window cursor only required one hand, and rated it as the most comfortable. However, we noticed that users generally had a poor sense of angular alignment, making the task of inferring real-world position difficult. For example, P2 found the window to be small; P4 and P7 found it hard to aim the camera at a correct angle.

> *Why is it not telling me anything when it sees text? (Because the window is not over the text) Oh... could they make that window bigger?* (P2)

> *It's like you have really narrow vision... You have to scan systematically left and then right.* (P7)

> *It's hard to tell if I'm actually tilting it or not... So technically speaking, holding it flat is tough.* (P4)

Finger Cursor

Finger cursor was not found to be successful for this task. Users thought the involvement of their finger was unnecessary.

> *That's a foolish way of doing things, I'm sorry to say this. It's just redundant. Have you ever seen a blind person pointing their finger at something?* (P11)

Many blind users had a tough time aligning both the objects and their finger in the camera's field-of-view. The "finger-in-view" earcon was generally not trusted, since the false positives were frustrating for many participants.

> *It's tougher because getting my finger in the camera view is what I'm finding to be hard... I would have to move both simultaneously.* (P4)

> *I think it's also very tiring... I would have to be super motivated to find out whether that was jam or not.* (P5)

Touch Cursor

Touch cursor was found to be the most difficult to use among the three. Only patient users with a systematic approach completed the task. A number of users wanted to freeze the live view. Even after discovering an object, users struggled to keep the device steady enough to locate it.

> *Actually, touching the screen is pretty good. It's a way to say, 'Hey, stop chattering, I'm looking for something right now.'... So in this case, if you were to freeze the image, that would be good.* (P11)

> *While I'm moving the phone around, the live view is changing, so I might miss the text on the screen.* (P8)

Sometimes, the user's grip would interfere with their success. For example, one participant never explored the bottom part of the screen. Participants also expressed the difficulty of looking for information on the screen when they had no notion of where the information was.

> *I think it would be much easier, if instead of me sliding my fingers around on the phone... that it would just read it out. Because I have no idea where the text is.* (P2)

> *It would be nice to get a little more feedback for hot and cold... or if the phone would vibrate the closer I get to something... because right now, it's just a tiny little screen in the dark!* (P3)

Task: Interpret Documents and Signs

In this task, we aim to answer the question, "Can users leverage window cursor, touch cursor, or finger cursor to interpret documents and signs?" The results suggest that touch cursor is the best among the three, with 5 most prefer, 2 neutral, and 2 least prefer.

Window Cursor

Window cursor was not found to be very effective for this task. Users mentioned that it was not clear whether the camera was capturing the whole page, or only a fraction of the page. The high density of entities recognized made the users' attempts to interpret documents ineffective.

You have to balance between density and truncating the text with the edge of the window, then you have to figure out how to scan it. (P3)

Moving along a row to gather table data was nearly impossible for most users, and they frequently had to guess the date/time based on ordering of audio output.

If it somehow can build a summary of what it thinks is important, that's good, but otherwise it has to read the whole thing. (P11)

Users were generally better at holding the device upright rather than perfectly flat. Users became fatigued by this task, and came up with creative ways to stabilize the device.

Finger Cursor

Users struggled not to occlude what they were attempting to read (especially if the document was not at chest height).

It seems to see my finger, but it's only reading small parts of stuff. (P8)

The desire to physically touch the flyers also caused users to stand too close to the bulletin. Users saw no value in pointing at specific sections of flyers because they did not know or care about page layout. Pointing at blank spaces and the edges of documents was a common mistake. It was not obvious that aiming for the top-center of a page might read the title.

If it takes too long to figure out, I'll probably just ask somebody... It's about getting things done, not about proving to the rest of the world that I can do this. (P5)

Touch cursor

Framing the task in a public setting caused users to worry about negative social judgment, and they reiterated the desire to capture a photo then walk away.

Is there a way to freeze the image? Then you could sit down and quietly explore... I would like to just take a screenshot and not take other people's time. (P5)

It's a little different in a coffee shop - people are going to wonder what the heck she's doing all hunched over. (P12)

Because text density was high, multiple entities would be selected and read out sequentially. Output was often garbled because of errors with OCR and text truncated by the camera's field-of-view. Shifts in users' body position compounded problems caused by latency. Such reasons add additional difficulty to reading documents and signs.

You have to move and hold still, it takes a lot of patience, because you move and you stop... then you move again, and you stop. It's like traffic: stop and go. (P1)

I find that moving my finger on the screen is throwing the orientation off, so because I'm fighting with it so much, I'm not noticing how much I'm moving the device. (P3)

Task: Read Labels and Enter Data on an Appliance

In this task, we aim to answer the question, "Can users leverage window cursor, touch cursor, or finger cursor to read labels and enter data?" The results suggest that finger cursor is the best among the three, with 6 most prefer and 2 neutral.

Window cursor

Users did not trust the mental model created only by proprioception (kinesthetic awareness).

Something blind people don't understand is how much or how little the camera can see, so we don't know how much we need to move the camera. (P12)

Latency caused scanning such a small area to be very difficult, and multiple buttons would often be read at once. Holding the device closer to the panel yielded the best results, but users who were unfamiliar with the device model did not know on which side the camera was located.

I'm having a hard time figuring out what the camera is capturing when I move it like that [waves device around in the air]. (P4)

You have to find the exact place, hold still, then find the button. (P7)

Finger Cursor

Unsurprisingly, users were the most confident and satisfied with this interaction for appliance usage.

I'm more sure I'm pressing the right button. (P7)

It's more tactile... Doing stuff on the screen, you still have to work through the screen to get to what you want, but if you're actually touching it, and it's telling you what you're interacting with, it's immediately much more useful. (P3)

It was the preferred method, but users still felt lost at times. Alignment and occlusion were the biggest hindrances to usability. Users did not instinctively realize that OCR could not recognize text directly beneath their finger.

How do I know if I'm covering the thing that I want to press? (P11)

On some panels, you can't really touch the buttons... since you might activate something. So that would be dangerous. And of course, because your finger will have to be below the actual control... I guess it's just something we would have to figure out... app should give you instructions... (P5)

Touch Cursor

Most users thought this method would be helpful in creating a mental blueprint of the panel.

Because I have to translate what's on the screen to what's on the board, a lot of that relational information is going to be lost..., but that would give a good overview of how the control panel is laid out. (P1)

The active window created by touching the screen was larger than the entities themselves, so the output was sometimes interpreted as contradictory and untrustworthy. Users also found it hard to capture the entire panel using the camera, and the mapping between screen and panel was not intuitive. Users

had slightly more success if they rested the edge of the device on the table to prevent unintentional shifts in the field-of-view.

> *Oh. Now I need to figure out where it is on the control panel? That is going to be totally impossible.* (P5)

> *I could find everything, but it was hard to know the relationship between the screen and the panel.* (P7)

Task: Learn About Surroundings

In this task, we aim to answer the question, "Can users leverage window cursor, touch cursor, or finger cursor to learn about their surroundings?" Exploring a 3D space was very slow because users needed to scan both vertically and horizontally. In addition to the original window cursor, we added a variation in which the window cursor shape is a vertical slit box in the middle of the screen (Figure 2b and Figure 3b). We also included the vertical window cursor as a fourth method in this task. Users preferred the vertical window cursor the most among the four methods. Finger cursor yielded the worst results because of latency, occlusion, and lack of confidence in the technique. Users hoped that the cursors could be used in combination with other environmental cues to provide additional information. We also observed that the "entity-in-view" earcon was less helpful for entities at a distance, and that holding the phone in one place was very hard.

> *The beep-beep-beep thing is not useful... It's telling me that there is text when I can't find the text. It seems to find text everywhere!* (P11)

> *Too slow and not enough feedback as to hot or cold, getting closer or away from whatever information that is causing the screen to beep... so I now have to hunt for that, and then once I find it, I have to interpret the position of the screen with what's in front of me, so... yea, I didn't like that interaction at all.* (P3)

Feedback From Low Vision Users

We observed high completion rates for low vision users. All users were able to complete the tasks using at least one method. Window cursor was the least preferred, touch cursor the most, and finger cursor had the highest potential. Reading text was inherently the most difficult task for low vision users. Both finger cursor and touch cursor were perceived as highly promising for reading documents and signs as well as for confirming control panel layout.

Latency became a more obvious issue for low vision users. Finger cursors are relatively intuitive for low vision users to pick up. In contrast to blind users, the feedback provided by the finger cursor is used for confirmation rather than information. Low vision users did not encounter issues related to aligning the camera, avoiding occlusion, or correctly targeting the field-of-view.

Feedback on the Visual Interface

Low vision users thought the overlay was too dark, or completely unnecessary. Users also wanted to change the size, shape, or scale of the window.

> *Oh, I found one... That's purely by chance though, because when I look through this, it's so much darker that I can't see anything at all.* (P9)

On-screen text was too small to be helpful, and it made entity-dense views more difficult to understand. The contrast of some entities was not sufficient. Users found that inconsistency in the assignment of colors to entities added to the visual noise. False positives with finger pointing were found to be problematic.

> *I can barely read that... The contrast of these make a big difference, so whatever the blue one is, I can't see it at all.* (P9)

> *There are so many boxes and so many colors that are overlapping one another. When I try to touch one, my finger is too fat, and I can't get the one [I want].* (P6)

Feedback on the Auditory Output

Users reported that the "entity-in-view" earcon was unnecessary since entity boxes appeared on the screen. "Finger-in-view" earcon was further redundant since the user could see the window, but not their finger. Users wanted more instruction on occlusion and framing.

> *The sounds are clear, but they're not helpful.* (P10)

> *It would be nice if it gave you a little more guidance... any time you were doing something that's not optimal.* (P9)

Feedback on the Physical Interaction

Window cursor was the least preferred method.

> *It's less efficient because there are multiple rectangles on the screen, and now I have to get the white rectangle over one of the colored boxes. It seems like it's redundant.* (P6)

> *It might be nice to zoom in a bit, especially if the words were smaller.* (P10)

> *I would like to be able to choose the whole paper, and not just one area.* (P9)

All of the low vision users thought pinching the screen to zoom might improve usability. Similar to blind users, low vision users also expected the number read out to be the number beneath their finger. False positives and latency while pointing their fingers were frustrating for users.

For touch cursor, users needed to be instructed to touch and hold, rather than just tap the boxes.

> *Until you instructed me to hold down longer than I'm used to, it was a bit confusing. I thought, just tap and let go.* (P6)

Because entities move on the screen as the camera moves, selection of the desired entity with touch cursor was difficult.

> *I can see the dates, but because these keep jumping around so much, I can't actually get them to be read out. For a person who's blind, that's totally unusable.* (P10)

Social Acceptability

Social acceptability emerges as a theme from the task sessions and interviews. Our participants also expressed concerns about safety when adopting new tools. Some users revealed that they will not adopt a tool that would make them appear significantly different from their peers.

> *Not to mention, you're going to look weird to other people. If a blind person randomly starts pointing their finger, people are going to think, 'Oh, what is this guy doing?'... You don't want to stand out from the crowd for making strange gestures.* (P5)

> *It's also a safety issue. I would be concerned if I were a blind person, and I was walking down the street, that someone would just grab [my phone] and steal it.* (P9)

> *People would stare...I don't know if I'm pointing at a person or what. They would be like, 'Why is this person pointing at me?'... I would be afraid of that.* (P11)

Key Takeaways

In the study we found that different cursor methods are more effective for different tasks. Window cursor works well for locating objects on larger surfaces, but does not work well for small and fine-grained tasks. Blind users generally liked this one the most, because it was the simplest to use, and required the least amount of coordination. Finger cursor works well for accessing appliance control panels, but does not work well for pointing at remote objects in the 3D space. Most blind users needed good instructions for this method, though finger cursor is the most intuitive for low vision users. Touch cursor works well for understanding the layout of a scene or document, but does not work well when mapping the on-screen locations is required to take actions on real-world objects (*e.g.*, grabbing an object, pushing a button). We also summarize the key concepts emerged from the task sessions and subsequent interviews:

- Familiarity with concepts of visual perception correlates to improved technique, greater patience, and higher success. Users who recently lost their sight find these interactions the most intuitive.
- Users believe the cursor methods are complementary and context-specific. Users believe a single task or scenario could benefit from multiple cursor methods used together.
- All cursor methods are potentially useful. But without guidance, they are too difficult to be valuable.
- Not all cursor methods should be triggered or detected automatically. Users enjoy a sense of control over their technology, and view cursors as an actively-triggered tool.
- The effectiveness of the interaction model depends on the form factor, and needs further validation. Most users expressed concerns for negative social judgment and personal safety while using the cursor methods.

DISCUSSION AND FUTURE WORK

The study revealed how blind people interacted with the three cursor methods. We identified system limitations, blind people's pain points and concerns. This informs future design that could better leverage and combine cursor-based interactions to support access to visual information in the real world.

Prerequisites for Cursor Usability

Blind participants found the system delay slowed them down and prevented them from interacting with the objects more intuitively. For blind users to interact with real world objects using cursor techniques in real time, recognition latency needs to be reduced, *e.g.* 0.1s according to approximate response time limits for instantaneous feedback[1].

The OCR recognizer's performance is quite poor for angled text, and often result in truncated and nonsensical output. In order for cursor interactions to work when the user is actively moving the device at different angles, robust recognition of angled and tilted text is necessary. Furthermore, other computer vision recognizers described earlier will be explored in future prototypes.

Consistent with prior work, providing additional feedback on the aiming of the camera would make the interactions more usable. One approach is to add edge detection for signs, documents, and panels, and provide feedback when the camera is not properly aligned. Alternatively, proper images can be automatically selected and used for recognition.

Auditory Feedback

Users found the earcons to be confusing. To improve the audio feedback for novice users, the "entity-in-view" earcon could be replaced with verbal hints, potentially with different styles of TTS or earcons of varied frequencies to indicate the distances of entities rather than the binary choice of playing or not playing an earcon. The "finger-in-view" earcon, from users' feedback, should be replaced with "finger-not-in-view," since warning the user when their finger is not properly aligned in the camera view might be more helpful. Furthermore, earcons should generally be more pleasant and expressive.

Cursor Interactions

Cursors help users actively seek information based on their needs, instead of passively receiving information of the entire environment. Future designs can enable blind people to have more control over what information they can get, or what they want to learn about their surroundings. For window cursor, the current "window" box could be replaced with a pinch-to-zoom or resizable window model, or by a "window" with a larger viewfinder frame. (We kept the cursor region the same for direct comparison in our study.)

Since users reported that finger cursor is less intuitive, we could provide a tutorial, and verify user's technique before enabling the cursor mode. Skin tone calibration could also be added to the finger detection algorithm to increase accuracy. To solve the occlusion problem of fingers, techniques such as keeping a memory of the entities in the scene, and using reference images for matching would help (*e.g.*, VizLens [8]).

For touch cursor, we found that it is not natural for the user to map the position of the entity on the screen to the location of

[1] https://www.nngroup.com/articles/response-times-3-important-limits/

the real world object, in order to take actions such as pushing a button or grabbing an object. However, this method does help with understanding the relative layout of different objects on the same document or scene. Therefore, it might be beneficial to apply touch cursor on still photographs or scans, and offer a touch-to-freeze or touch-to-save model. For example, after the user scans a document, they could use touch cursor to navigate the document. Furthermore, the user could take a picture or a panorama of a scene, and use touch cursor to explore the layout of different objects, similar to RegionSpeak [28].

For many tasks, users' ideal output would be an intelligent and well-formed summary, which they could probe for more details if needed, as mentioned by P1, P3, P9, P11, and P12. This points to the need of providing both target information and context-relevant information at the same time [23]. We could explore adding a search feature in the tool that enables users to look for specific information within a larger context.

Social Acceptability

An important theme in participant comments was social acceptability. The findings suggest that when designing tools for blind and low vision users, in addition to technical feasibility and efficiency, social acceptability is also a key factor. This was not emphasized in prior work. We see research or commercial systems fail to consider social acceptability when designing interaction methods. For example, OrCam [18] users point their finger at or on an object they want to recognize, then pull it away for a picture to be snapped, which may not be appropriate for targeting remote objects in public. In future work, we need to further investigate this adoption issue, and ensure blind and low vision users do not feel awkward, obtrusive, or self-conscious.

Combination of Multiple Cursor Methods

From the study, we realized that for many use cases, applying a combination of multiple modes might be more helpful for the user. For example, when trying to access an appliance control panel, they could first scan the control panel, and use touch cursor to explore and familiarize themselves with the layout of the interface, then use the finger cursor mode to access the buttons. When trying to find an object on a table, it would be helpful to first get an idea of the available objects and their relative layout with touch cursor, then use window cursor to locate a specific one. When trying to read a bulletin board, it would be helpful to first find the specific poster they are interested in with window/finger cursor, then scan it and use touch cursor to explore the document. Finally, when walking in an unfamiliar environment, it would be helpful to first know an overview of the things around them, then use vertical window cursor to guide them to the direction of a specific one. In this case, they could wear the device in a lanyard so they do not need to hold it.

If applying a combination of multiple cursor modes would be more helpful, having the users manually switch between them might be cumbersome, especially when they are wearing the device and their hands are busy holding a cane or guide dog. Therefore, automatically switching between cursor-based interaction modes could be interesting to explore. For example, as soon as they take the device out of the lanyard and hold it up with their hand, window cursor could be automatically selected. Then, if they show their hand in the field of view of the camera, finger cursor could be launched, if the user touches the screen or start dragging their finger across the screen, touch cursor could be triggered.

CONCLUSIONS

In this paper, we implemented three cursor-based interactions that have been explored in prior work individually, to help blind users access targeted information from visual scenes. We conducted a thorough study to evaluate and compare the three cursors across four tasks that are representative of daily routines. The study reveals that different cursor methods are effective for different tasks. More specifically, we found that window cursor works well for locating objects on large surfaces; finger cursor works well for accessing appliance control panels; and touch cursor works well for understanding spatial layouts. A combination of multiple techniques will likely be best for supporting a variety of everyday tasks for blind users.

ACKNOWLEDGEMENTS

We thank the Google accessibility engineering team for their support, Brian Carlo Chuapoco for his effort in participant recruiting, to all our study participants for their time, and the reviewers for their valuable feedback and suggestions.

REFERENCES

1. Aipoly 2018. Aipoly. http://aipoly.com. (2018).
2. Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. 2010a. VizWiz: Nearly Real-time Answers to Visual Questions. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 333–342. DOI: http://dx.doi.org/10.1145/1866029.1866080
3. Jeffrey P Bigham, Chandrika Jayant, Andrew Miller, Brandyn White, and Tom Yeh. 2010b. VizWiz:: LocateIt-enabling blind people to locate objects in their environment. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 65–72.
4. Meera M Blattner, Denise A Sumikawa, and Robert M Greenberg. 1989. Earcons and icons: Their structure and common design principles. *Human–Computer Interaction* 4, 1 (1989), 11–44.
5. Glenn A Bowen. 2008. Naturalistic inquiry and the saturation concept: a research note. *Qualitative research* 8, 1 (2008), 137–152.
6. Wikipedia contributors. 2018. Optical character recognition — Wikipedia, The Free Encyclopedia. (2018). https://en.wikipedia.org/w/index.php?title=Optical_character_recognition&oldid=847751695 [Online; accessed 27-June-2018].

7. Giovanni Fusco, Ender Tekin, Richard E. Ladner, and James M. Coughlan. 2014. Using Computer Vision to Access Appliance Displays. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '14)*. ACM, New York, NY, USA, 281–282. DOI: http://dx.doi.org/10.1145/2661334.2661404

8. Anhong Guo, Xiang 'Anthony' Chen, Haoran Qi, Samuel White, Suman Ghosh, Chieko Asakawa, and Jeffrey P. Bigham. 2016. VizLens: A robust and interactive screen reader for interfaces in the real world. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 651–664. DOI: http://dx.doi.org/10.1145/2984511.2984518

9. Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P. Bigham. 2011. Supporting Blind Photography. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '11)*. ACM, New York, NY, USA, 203–210. DOI: http://dx.doi.org/10.1145/2049536.2049573

10. 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 (Assets '08)*. ACM, New York, NY, USA, 73–80. DOI: http://dx.doi.org/10.1145/1414471.1414487

11. Shaun K. Kane, Brian Frey, and Jacob O. Wobbrock. 2013. Access Lens: A Gesture-based Screen Reader for Real-world Documents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 347–350. DOI: http://dx.doi.org/10.1145/2470654.2470704

12. KNFB Reader 2018. KNFB Reader. http://www.knfbreader.com/. (2018).

13. LookTel Money Reader 2018. LookTel Money Reader. http://www.looktel.com/moneyreader. (2018).

14. LookTel Recognizer 2018. LookTel Recognizer. http://www.looktel.com/recognizer. (2018).

15. Roberto Manduchi and James M. Coughlan. 2014. The Last Meter: Blind Visual Guidance to a Target. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3113–3122. DOI: http://dx.doi.org/10.1145/2556288.2557328

16. T. Morris, P. Blenkhorn, L. Crossey, Q. Ngo, M. Ross, D. Werner, and C. Wong. 2006. Clearspeech: A Display Reader for the Visually Handicapped. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14, 4 (Dec 2006), 492–500. DOI: http://dx.doi.org/10.1109/TNSRE.2006.881538

17. Suranga Nanayakkara, Roy Shilkrot, Kian Peen Yeo, and Pattie Maes. 2013. EyeRing: A Finger-worn Input Device for Seamless Interactions with Our Surroundings. In *Proceedings of the 4th Augmented Human International Conference (AH '13)*. ACM, New York, NY, USA, 13–20. DOI:http://dx.doi.org/10.1145/2459236.2459240

18. OrCam 2018. OrCam. http://www.orcam.com. (2018).

19. Seeing AI 2018. Seeing AI. https://www.microsoft.com/en-us/seeing-ai/. (2018).

20. Lei Shi, Yuhang Zhao, and Shiri Azenkot. 2017. Markit and Talkit: A Low-Barrier Toolkit to Augment 3D Printed Models with Audio Annotations. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 493–506. DOI: http://dx.doi.org/10.1145/3126594.3126650

21. Roy Shilkrot, Jochen Huber, Connie Liu, Pattie Maes, and Suranga Chandima Nanayakkara. 2014. FingerReader: A Wearable Device to Support Text Reading on the Go. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 2359–2364. DOI: http://dx.doi.org/10.1145/2559206.2581220

22. Kristen Shinohara and Josh Tenenberg. 2007. Observing Sara: A Case Study of a Blind Person's Interactions with Technology. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '07)*. ACM, New York, NY, USA, 171–178. DOI: http://dx.doi.org/10.1145/1296843.1296873

23. B. Shneiderman. 1996. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*. 336–343. DOI: http://dx.doi.org/10.1109/VL.1996.545307

24. Ender Tekin, James M. Coughlan, and Huiying Shen. 2011. Real-time Detection and Reading of LED/LCD Displays for Visually Impaired Persons. In *Proceedings of the 2011 IEEE Workshop on Applications of Computer Vision (WACV) (WACV '11)*. IEEE Computer Society, Washington, DC, USA, 491–496. DOI: http://dx.doi.org/10.1109/WACV.2011.5711544

25. Marynel Vázquez and Aaron Steinfeld. 2014. An Assisted Photography Framework to Help Visually Impaired Users Properly Aim a Camera. *ACM Trans. Comput.-Hum. Interact.* 21, 5, Article 25 (Nov. 2014), 29 pages. DOI:http://dx.doi.org/10.1145/2651380

26. Samuel White, Hanjie Ji, and Jeffrey P. Bigham. 2010. EasySnap: Real-time Audio Feedback for Blind Photography. In *Adjunct Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 409–410. DOI: http://dx.doi.org/10.1145/1866218.1866244

27. Yu Zhong, Pierre J. Garrigues, and Jeffrey P. Bigham. 2013. Real Time Object Scanning Using a Mobile Phone and Cloud-based Visual Search Engine. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13)*. ACM, New York, NY, USA, Article 20, 8 pages. DOI: http://dx.doi.org/10.1145/2513383.2513443

28. Yu Zhong, Walter S. Lasecki, Erin Brady, and Jeffrey P. Bigham. 2015. RegionSpeak: Quick Comprehensive Spatial Descriptions of Complex Images for Blind Users. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2353–2362. DOI: http://dx.doi.org/10.1145/2702123.2702437