Toucha11y: Making Inaccessible Public Touchscreens Accessible

Jiasheng Li
jsli@umd.edu
University of Maryland
College Park, Maryland, USA

Zeyu Yan
zeyuy@umd.edu
University of Maryland
College Park, Maryland, USA

Arush Shah
arushshah1@gmail.com
University of Maryland
College Park, Maryland, USA

Jonathan Lazar
jlazar@umd.edu
University of Maryland
College Park, Maryland, USA

Huaishu Peng
huaishu@umd.edu
University of Maryland
College Park, Maryland, USA

ABSTRACT

Despite their growing popularity, many public kiosks with touchscreens are inaccessible to blind people. Toucha11y is a working prototype that allows blind users to use existing inaccessible touchscreen kiosks independently and with little effort. Toucha11y consists of a mechanical bot that can be instrumented to an arbitrary touchscreen kiosk by a blind user and a companion app on their smartphone. The bot, once attached to a touchscreen, will recognize its content, retrieve the corresponding information from a database, and render it on the user's smartphone. As a result, a blind person can use the smartphone's built-in accessibility features to access content and make selections. The mechanical bot will detect and activate the corresponding touchscreen interface. We present the system design of Toucha11y along with a series of technical evaluations. Through a user study, we found out that Toucha11y could help blind users operate inaccessible touchscreen devices.

CCS CONCEPTS

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

KEYWORDS

accessibility, touchscreen appliances, robotic, visual impairments

1 INTRODUCTION

From self-service food in restaurants to ID renewal in the Department of Motor Vehicles (DMV), touchscreen kiosks have been increasingly popular in workplaces and public access areas. The emergence of the COVID-19 pandemic further accelerates the adoption

Figure 1: Toucha11y overview. a) The Toucha11y bot can be placed on top of an inaccessible touchscreen kiosk to activate its interface. b) A blind user can access and use the kiosk from their personal smartphone.

of touchscreen devices [58], as they provide customers with an autonomous experience while promoting social distancing. Unfortunately, the widespread use of touchscreen technology on kiosks can be challenging for blind people. Being primarily visual and nonhaptic, many touchscreen devices—especially those in public—are not accessible to them, preventing blind users from performing tasks independently and potentially introducing feelings of embarrassment, as discussed by Kane et al. [24].

Much effort has recently been made to improve the accessibility of public touchscreen devices. Several recent lawsuits in the United States, for example, have compelled large corporations to make their public touchscreen devices accessible [38]. The Americans with Disabilities Act (ADA) has also required that public kiosks (such as ATMs and movie rental kiosks) be accessible to blind users [2]. Indeed, despite the slow pace, we are seeing an increasing number of public touchscreen devices outfitted with accessibility features [13]. Unfortunately, because the guidelines for designing accessible kiosks are not clearly defined, the majority of public touchscreens are still not useful to blind users [30]. Even devices with touchscreens and physical keypads, for example, may not have any audio output [30]; others with audio output may simply speak the information publicly, ignoring the privacy of blind users. Moreover, a significant number of inaccessible touchscreens have already been deployed worldwide. Waiting for device manufacturers to update all of them will not solve today's accessibility problems.

To make those already-implemented public devices accessible to blind users, researchers in the field of HCI have proposed different methods to augment or retrofit the touchscreens. For example, Flexible Access System [48], Touchplates [25], and Facade [18] propose enhancing existing touchscreens with (3D-printable) physical buttons and custom tactile overlays. Although promising, it is impractical for blind users to install custom guidance on arbitrary public touchscreens they encounter in their daily lives. VizLens [17] proposes a crowdsourcing solution in which photos of a touchscreen interface taken by blind users are first labeled by crowd workers and then used to guide blind users to operate the touchscreen. This method works with static touchscreens (such as those on microwave ovens), but it can be difficult with touchscreens with changeable content. To work with dynamic touchscreen interfaces, Statelens [19] proposes reverse engineering their underlying state diagrams. It then generates instructions for blind users to use the touchscreens in conjunction with a custom 3D-printed accessory. Statelens allows blind users to explore arbitrary public touchscreens independently, but in order for the step-wise guidance to work, blind users must hold their phone with one hand and keep its camera focused on the touchscreen while exploring the screen interface with the other. In such cases, bimanual operations can be difficult for some blind users. Besides, since the content and layout of touchscreen interfaces can vary, physically learning and exploring a touchscreen interface can be time-consuming and laborious.

Building upon previous research, we present Toucha11y[1] , a working prototype that aims to enable blind users to access arbitrary public touchscreen devices independently and with little effort. The key to Toucha11y is to bridge the gap between blind users and a public touchscreen device with a set of hardware and software tools, allowing them to explore touchscreen content from their familiar smartphone devices without having to deal with the unfamiliar, inaccessible public touchscreen directly. To make the bridge work, Toucha11y's hardware—a small mechanical bot—must

be placed on top of a public touchscreen device by a blind user (Figure 1a). Once placed, the bot's onboard camera will photograph the screen, with its corresponding interface (which can be generated through crowdsourcing and reverse engineering [19]) sent to the user's smartphone. The blind user can freely explore and select contents using the smartphone's built-in accessibility features such as Apple's VoiceOver and Android's TalkBack [7, 16] (Figure 1b). These selections will be sent back to the bot, which will physically register the corresponding touch event for the blind user using an extendable reel. With Toucha11y, a blind user only needs to interact with an inaccessible touchscreen once, when placing the bot on the screen. Because the remainder of the interactions occurs on the user's smartphone, a blind user is not required to learn how to use a new touchscreen interface every time a new device is encountered. Toucha11y can also help alleviate privacy concerns by allowing blind users to use their personal devices for input and voice output, which they likely already have configured for privacy (with earphones or other options).

Toucha11y's design is inspired by a list of previous works as well as an interview with nine blind participants. In the following sections, we will first present the interview findings and the distilled design guidance. We will then detail its system implementation, including the key features of the mechanical bot, the smartphone interface, and the companion back-end server. Through a series of technical evaluations, we demonstrate that the Toucha11y bot can accurately recognize the content of a touchscreen and its relative location. The extendable reel can also reach target locations with high accuracy. A user study with seven blind participants shows that Toucha11y can provide blind users with a solution to independently use an inaccessible touchscreen kiosk interface.

In summary, our paper contributes to: 1) the investigation of blind users' practices and challenges in using public touchscreen devices; 2) Toucha11y, a working prototype that makes inaccessible touchscreen devices accessible to blind users; and 3) a user study to evaluate the tool.

2 RELATED WORK

Our work builds upon the notions of 1) touchscreen accessibility, 2) personal assistive devices, and 3) accessibility with computer vision.

2.1 Touchscreen Accessibility

Over the last two decades, the widespread use of touchscreens on mobile phones and tablets has raised serious concerns about their accessibility for blind and low-vision users. In response, numerous efforts have been made in both academia and industry to increase their accessibility. For example, Talking Fingertip Technique [49], Slide Rule [22], among others [47, 55, 56], propose using (multi-)touch gestures to control touchscreens non-visually, which influenced the design of VoiceOver [7] and TalkBack [16], that are now widely adopted in modern smartphone devices. Reports have shown that smartphones are now frequently used among blind users and are even replacing traditional solutions [34].

Touchscreens are also prevalent outside of personal computing devices today. Unfortunately, the growing number of public

---

[1]Toucha11y is a combination of the words "touch" and "accessibility." The "11" refers to the 11 letters in "accessibility" between "a" and "y." See more at https://www.a11yproject.com/

kiosks with touchscreens, such as those in hospitals, airports, markets, and restaurants, are not as accessible as smartphones. Despite efforts to improve their accessibility (e.g., ADA laws in the United States [2]), a large proportion of existing kiosks with touchscreens are inaccessible to blind people [30]. In order to prevent their exclusion, researchers have proposed techniques to make existing public touchscreen devices accessible. One approach is to provide haptics, physical buttons, or tactile displays to existing touchscreens [17, 18, 25, 48]. While additional haptics can help blind users understand a touchscreen interface, it is impractical for blind people to install bespoke assistance on random public touchscreens for touch guidance. Rather than retrofitting the touchscreen, Statelens [19] proposes to use computer vision and crowdsourcing to reverse engineer the underlying state diagrams of a touchscreen interface. From these state diagrams, step-by-step instructions can be generated to assist blind individuals in using touchscreens with a 3D-printed accessory.

Building upon the reverse engineering approach, Toucha11y assumes that labeled touchscreen interfaces can be retrieved from a shared repository or database. Unlike Statelens, Toucha11y moves the interface exploration from the inaccessible touchscreen to the users' phones. This way, blind users don't have to learn how to use each new public kiosk they encounter.

2.2 Personal Assistive Devices

While not widely used yet [36, 37], personal assistive devices [50, 54] have the potential to improve real-world accessibility for blind users in a variety of contexts, from providing travel and navigation aids [11, 15, 50, 53] to promoting individual creativity [12, 26, 32]. Personal assistive devices have also been utilized to support situations such as shopping in order to promote independence. Shoptalk [42], for instance, aims to improve the shopping experience of blind users with handheld barcode scanners. Foresee [57] demonstrates how a head-mounted vision solution can magnify real-world objects to assist low-vision users in the grocery store.

Because of their embodiment possibilities and potential social presence, robots in various forms have been investigated as a means of assistive technology in addition to wearables or handheld devices. Mobile robots, like wearable devices, have been used for navigation and road guidance [27–29]. Drones have also been investigated for use in guiding blind runners and assisting with indoor navigation [5, 8].

Toucha11y is also a personal assistive robot in that it is designed to be carried by the blind user. Unlike the previously mentioned research focusing on navigation and travel aids, Toucha11y proposes to automate existing and inaccessible touchscreen activation for blind users, which, to the best of our knowledge, has not previously been investigated.

2.3 Accessibility with Computer Vision

A large number of assistive technologies are based on computer vision. For example, using optical character recognition (OCR), several systems (such as the KNFB Reader [43]) have been developed to assist blind people in reading visual text. Camera-based solutions, such as those attached to a table [23], worn on the finger [41, 45],

or held in the hand [39], are proposed to recognize text in physical documents and allow a blind person to hear and interact with them.

Recent advancements in deep learning have enabled commercial solutions such as Seeing AI [35] and Aipoly [4] to apply general object recognition to identify a variety of things, including currencies, text, documents, and people, among others. One remaining challenge is automatic labeling and recognizing photos captured by blind users [20], which can be crucial for crowdsourcing-based assistive systems (e.g., Vizwiz [10]) when crowd workers are not immediately available.

Toucha11y also makes use of computer vision for two purposes. Toucha11y uses the classic SIFT algorithm [33] to detect the key points from the photos taken by the onboard camera. It then matches the photos to the stocked touchscreen interfaces using FLANN [40], and utilizes the positions of these photos to compute the bot's actual location on the screen's X-Y coordinates.

3 FORMATIVE STUDY

To gain insights about blind users' experience with touchscreen-based kiosks (if any) and to identify key challenges, we conducted semi-structured interviews with nine blind individuals. Each interview was approximately thirty minutes long and was audio-transcribed. The design of Toucha11y was guided by the findings from the interviews and previous literature.

3.1 Findings

3.1.1 Common practice. While all interviewees reported using touchscreen-based personal devices (e.g., iPhones and Android smartphones) on a daily basis, none used public touchscreen-based kiosks regularly. Six of the interviewees noted that they had never used a public kiosk. For the remaining three, they reported a one-time experience of trying a kiosk on different occasions (to withdraw cash from an ATM, to make an appointment for a lab test in a clinic, and to renew an ID at the DMV), but because the experience was so negative, they no longer use kiosks by themselves. For all the participants we interviewed, when they needed to use a touchscreen-based kiosk, they either sought assistance from sighted people or chose not to use the device at all.

3.1.2 Barriers to using kiosks. Two main challenges were identified as preventing blind interviewees from trying to use a public kiosk. First, participants noted that figuring out whether a public kiosk was accessible or not was challenging. Interviewees claimed that they were aware that kiosks were growing in popularity and that they had encountered them throughout their lives. Four participants also stated that they were aware that ADA regulations require that kiosks be accessible. However, with a huge percentage of existing kiosks being inaccessible, *"the common sense for blind people when we come across a device with a touchscreen is that it doesn't work for us."* Thus, blind users who wish to use one must make additional efforts to first figure out whether or not it is usable.

Second, even kiosks that claimed to be accessible could be difficult to learn and use. One participant described their arduous experience withdrawing cash from an ATM kiosk. *"I was told that the ATM was accessible, so I wanted to give it a chance... I found the ATM, looked for the headphone jack on the machine, and plugged my headphones into it. Although the bank claimed that I could navigate*

the screen to check my balance or withdraw money, there was no clear instruction on how to do it after I plugged in the headphones." The interviewee ended up spending an hour figuring out how to use the touchscreen ATM on their own because no one was in the lobby to assist them. *"It wasn't a very good experience, and that didn't work out. So at last, I gave up and left the bank."*

Because blind participants had either experienced or were aware that many touchscreen-based kiosks were not accessible, they concluded that learning to use them would not pay off. As one participant put it, *"if a sighted person could use a touchscreen in 30 seconds to finish the job, but I need to spend 10 minutes learning and using it, why should I?"*

3.1.3 Privacy concerns. Since the use of kiosks was impractical for many blind users, the only viable alternative was to seek assistance from sighted individuals. However, when a blind individual is not accompanied by a friend or family member, seeking assistance from strangers could expose them to the risk of disclosing sensitive personal information. One participant stated, *"There was a kiosk to let people register in the Social Security Office, but the machine wasn't accessible to me. I had to find someone on-site to help me, and I needed to tell them my social security number, which I didn't wish to do."* Other participants echoed this sentiment, *"If I use a touchscreen-based ATM, will I even let others know my password? No, that isn't what I want."*

3.2 Design Considerations

The interview findings inform the three design considerations as stated below.

(1) *Simplify the interaction.* As it remains difficult for blind users to figure out and learn how to use an arbitrary public kiosk due to the inconsistency of (existing) accessibility features, we will focus on simplifying the learning and interaction processes. Knowing that many blind users are active smartphone users, we aim to offload the interaction with a public kiosk to the user's personal smartphone device. The actual touch event on a touchscreen can be delegated to a robot that excels at repeatable and accurate touch actions.

(2) *Protect privacy and promote autonomy.* Our solution shall protect the privacy of blind users and enable them to independently use touchscreen devices. For example, our solution shall allow blind people to enter their passwords or personal information without the assistance of sighted people.

(3) *Save time.* Blind people may require more time than sighted people to interact with a touchscreen device they are unfamiliar with. Our solution shall be time-efficient.

4 TOUCHA11Y WALKTHROUGH

Toucha11y is a personal assistive device that enables blind people to use existing, inaccessible public touchscreen devices independently and with minimal effort. To better explain how Toucha11y works, in this section, we will walk through the use of Toucha11y in an example scenario where Alex—a blind user of Toucha11y—is attempting to order a bubble tea through an inaccessible public kiosk.

Alex is a blind user who enjoys bubble tea. Alex's favorite bubble tea shop recently began using a self-service kiosk for all food orders.

Figure 2: Using Toucha11y, Alex can access the touchscreen content from a smartphone.

Unfortunately, despite being brand new, the kiosk is touchscreen-only with no accessibility features, which Alex discovered only after arriving at the store.

Although not ideal, with Toucha11y, Alex is able to order a drink at the self-serving kiosk (Figure 2). Alex first approaches the kiosk, locates its screen, and then attaches the Toucha11y bot to it. Confirming that the bot is securely in place, Alex launches the Toucha11y app on the smartphone. The bot begins to rotate and photograph the screen, allowing it to recognize the touchscreen context as well as its own location in relation to the touchscreen; at the same time, the mobile app retrieves and displays the corresponding menu of the bubble tea shop on Alex's smartphone. Instead of physically touching the inaccessible kiosk screen, Alex can use the phone to browse the menu using the smartphone's built-in screen reader features.

Alex selects "Avocado Tea" and presses the "Add" button on the smartphone. The bot then activates the avocado tea button on the physical touchscreen, bringing the ordering process to the next step, tea customization. At the same time, Toucha11y the app prompts Alex with the same customization options. Alex makes their choice and presses the "Add to Cart" button on the phone. Meanwhile, the Toucha11y bot performs these actions on the physical touchscreen, completing Alex's order.

5 TOUCHA11Y

Figure 3 shows the Toucha11y architecture, which consists of a mechanical bot that physically registers touch events for a blind user, a smartphone app that the user can interact with, and a back-end server that bridges users' input and the bot's actuation. Note that since previous work, such as Statelens [19], has already detailed how to reverse engineer and label an arbitrary touchscreen interface, we assume that the touchscreen interface information can be directly retrieved from the server. Built upon, Toucha11y's main focus is a tangible solution that allows blind users to access public touchscreens with a simple interaction model, less concern for privacy, and independence.
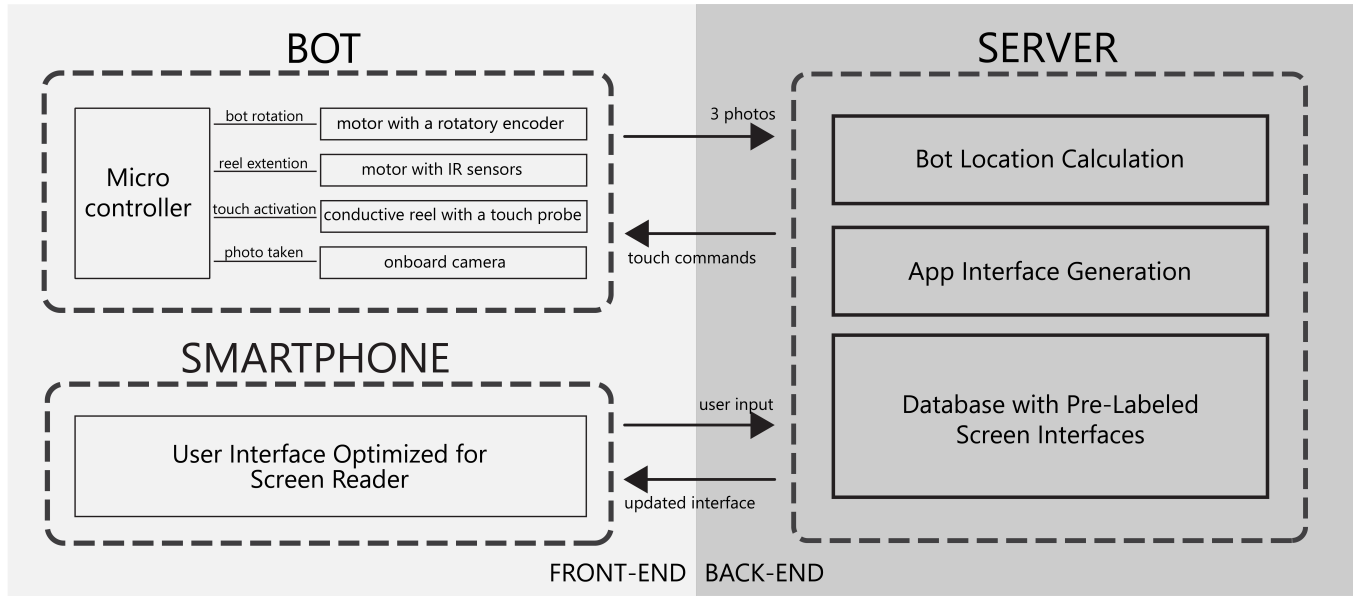
Below we detail each of the Toucha11y system components.

Figure 3: Toucha11y system architecture.

5.1 The Toucha11y Bot

Figure 4a is an exploded view of the bot design—the actuator of the Toucha11y system that physically registers touch events for a blind user. To accomplish this, the bot must be able to: 1) fix to a touchscreen surface; 2) locate itself on the touchscreen; and 3) register touch events according to the user's input.

5.1.1 Fix to a touchscreen surface. The first step to using Toucha11y is to fix the bot to a touchscreen kiosk. To ease the anchoring process for blind users, the base of the Toucha11y bot is equipped with three 19 mm diameter suction cups spaced 120° apart (Figure 5b). Three suction cups will guarantee the bot's overall stability while it physically activates the screen; the redundancy in the number of suction cups is to ensure that the device can still stay on the screen even if one of the suction cups fails.

5.1.2 Photograph the touchscreen for localization. Once the bot is fixed on the touchscreen, it needs to recognize its precise placement with respect to the screen's coordinates in order to register touch events. Toucha11y bot accomplishes this by taking three consecutive photos of the screen interface beneath (each with a 30° gap), which, after each shot, are uploaded to the back-end server. Since the bot's microcontroller, the Raspberry Pi Zero W, is not suitable for heavy computation, the on-screen localization is done on the server using off-the-shelf computer vision algorithms SIFT [33] and FLANN [40], which we will briefly explain in Section 5.3.2.

To take photos of the screen, the bot is equipped with a Pi V2 camera (8 megapixels, 1080p), mounted 70 mm above the bot's base and angled 45° downward (Figure 4b). This configuration is to ensure that each photo captured by the camera can cover a large screen area, in this case, 120 mm^2. Currently, it takes the bot four seconds to capture all three photos of the screen.
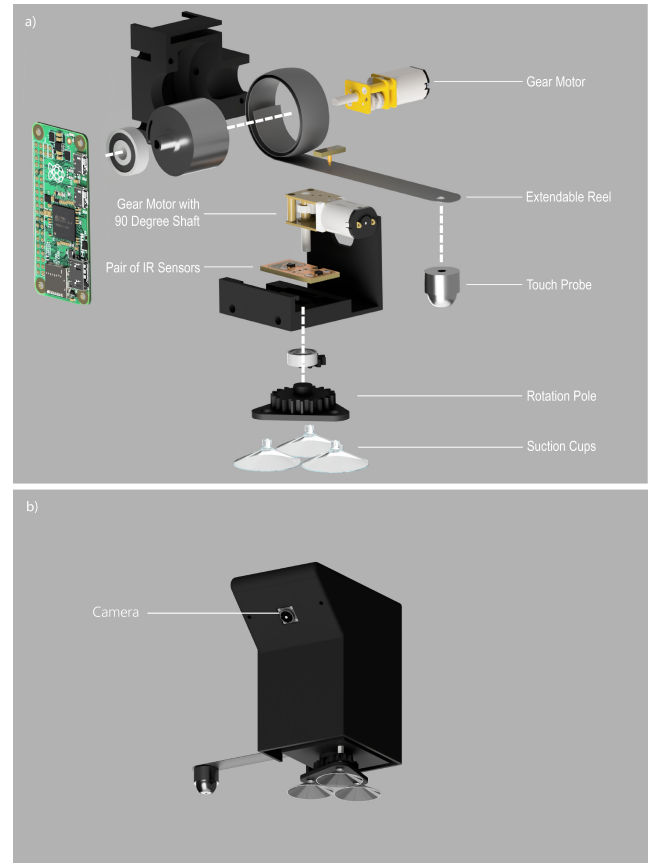

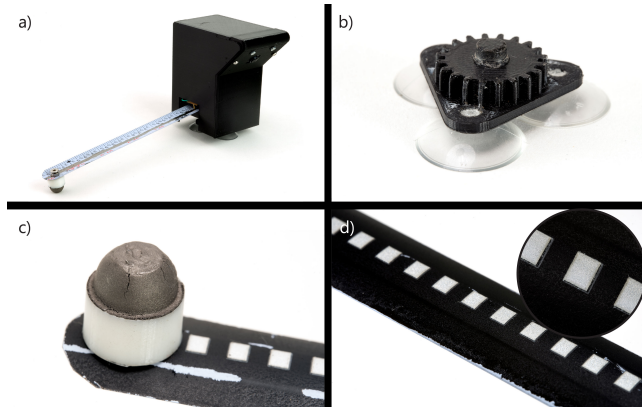**Figure 4: Toucha11y assembly illustration.**

Figure 5: a) The Toucha11y bot. b) The bot base with three suction cups. c) A 3D-printed touch probe coated with conductive material. d) Extendable reel made from stainless steel measuring tape with black backing and reflective white stripes.

5.1.3 Touch event registration.

5.1.3 Touch event registration. After locating itself on a touchscreen, the bot is ready to register touch events mechanically. The bot design employs two-dimensional polar coordinates where the main body of the bot rotates along the origin and an extendable reel can change length along the polar axis (Figure 4b). Such design makes it possible for the extendable reel to reach virtual screen buttons that are far from the bot's body while remaining completely hidden when not in use, thereby reducing the overall size of the bot.

The extendable reel design is inspired by [46]. The reel is made from a portion of a stainless steel measuring tape (Figure 5a). Through experiments, we found that the stainless steel reel is both rigid and lightweight, allowing it to remain straight even when extended over a great distance (up to 700 mm with our current prototype). The far end of the extendable reel is installed with a 3D-printed touch probe (coated with conductive paint, MG Chemicals MG841AR) facing the touchscreen surface (Figure 5c). When a virtual button on a touchscreen needs to be activated, the reel first extends out from the bot body at an angle of about 5° pointing downward (Figure 5a). Once the touch probe reaches the target, a touch event signal from the bot's microcontroller is transmitted through the stainless steel reel to the touch probe, which then activates the touchscreen interface. Note that the slight tilting angle of the extendable reel is to ensure that the touch probe always has secure contact against the touchscreen across the entire area. Also, as the touch mechanism is activated electrically, a touch event will only be triggered once the probe arrives at the prospective location.

To ensure that the length of the extrusion is accurate, the back side of the reel is painted black with reflective white stripes evenly spaced (at a pitch of 2.5 mm), which are detected by a pair of IR sensors for length counting (Figure 5d). The maximum reach of the current prototype is 700 mm, which is enough to encircle a 40-inch touchscreen device if placed at its center. The extrusion speed of the reel is 25 mm/s; the rotation speed is 5 rpm.

5.1.4 Implementation. The bot prototype is self-contained, with an overall size of 50 mm by 70 mm by 93 mm and a weight of 162.53 g.
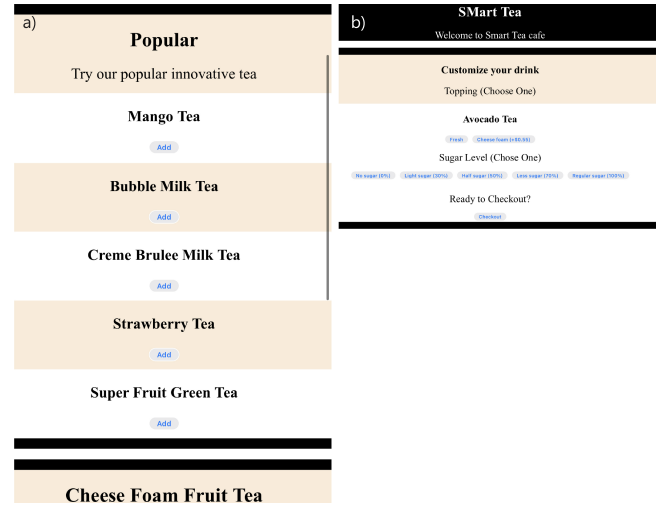

Figure 6: An example of the smartphone interface for blind users.

The bot is equipped with a Raspberry Pi Zero W as the microcontroller, two N20 gearmotors to drive the rotation and the reel extension, and a 7.4 V, 260 mah LiPo battery as the power source. Given that the controller only accepts 5 V as input voltage, we use an MP1584 DC-DC step-down converter to regulate the voltage. Additionally, the battery powers the two motors through a dual H-bridge DRV8833. The pole rotation motor is equipped with a magnetic rotary encoder and a 90° shaft that connects to the suction cup base. The motor of the extension reel is positioned orthogonally above (Figure 4a). Two IR sensors (SHARP GP2S60) are positioned beneath the extendable reel to measure extension length. The rotation angle of the bot and the length of the extendable reel are both computed using a custom PID implementation.

5.2 The Smartphone Interface

The main goal of the smartphone interface is to allow blind users to access and interact with the content of a public kiosk using their own equipment, which not only has accessibility features built-in but can also mitigate privacy concerns.

In the current implementation, the HTML-based smartphone interface is automatically generated on the server and pushed to the user's phone (see Section 5.3.3 for interface generation). Figure 6 showcases one example of a bubble tea menu being converted to a smartphone interface. Note that all touchable elements are marked up in HTML according to WCAG standards [52] so that they can be verbalized by a screen reader.

5.3 The Server

While the back-end server is not directly accessed by end users, it has three key functions that bridge the bot actuation and user input. In particular, the Toucha11y server stores the pre-labeled touchscreen interface information in a database, computes the placement of the bot, and generates the smartphone interface for the user. It also sends touch commands to the bot based on the user's selection
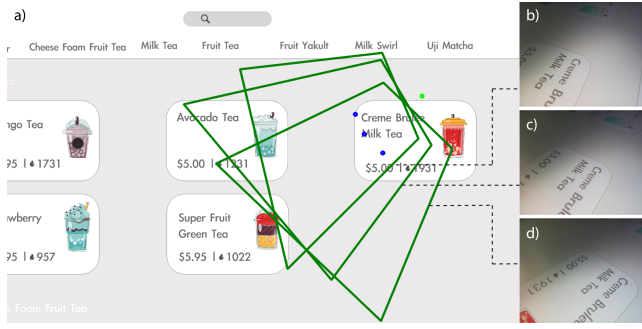
Figure 7: a) The green boxes represent the perspective-transformed area covered by the photos taken by the on-board camera, shown as b), c), and d). The green dot shows the estimated location computed from the three blue dots, which are the perspective-transformed points corresponding to the photos' center coordinates.

Figure 8: Samples of touchscreens' interfaces.

from their smartphone, and updates the phone interface whenever the touchscreen is updated (Figure 3).

Flask [44] is used to implement the server, which is deployed on the Heroku online cloud platform. The server communicates with both the bot and the user's smartphone wirelessly.

5.3.1 Database. A simple, customized database is hosted on the server and stores the essential information of the public touchscreen interfaces. Specifically, for each instance of a touchscreen interface, three types of data are stored, including all the text contents on the interface, their click-ability, and their location coordinates on the screens. The first two are used to generate the smartphone interface, and the location coordinates are sent to the bot for touch activation. Additionally, one image per interface is stored in the database for computing the bot's location. Note that the interfaces in the database are manually labeled for the current prototype, but we assume that they can be labeled via crowdsourcing if deployed in the future, as discussed in [19].

5.3.2 Location computation. The server determines the location of the bot based on the three photos taken by the bot's onboard camera and the stored kiosk interface images in its database. The computation can be broken down into three steps: finding the matching interface image, computing the locations of the camera photos in the interface image coordinates, and then triangulating the bot's location.

To determine which touchscreen kiosk the bot is placed on and its corresponding interface image in the database, the server first uses SIFT [33] to generate a list of key points (represented as vectors) in the camera photos. These key points are then compared to those of the pre-labeled interface images in the database. This is achieved using FLANN [40] by finding the nearest neighbors of the interface image's key points that match those of the camera photos; the one with the highest number of matching key points is regarded as the touchscreen interface on which the bot is placed on. Once the interface image has been identified, the server uses a perspective transformation to compute the locations of all three camera photos within its coordinates. For example, the three green

boxes in Figure 7 represent the transformed area of the three camera photos, whereas the three blue dots mark the geographical center of the camera photos after the transformation. A circle that traverses the transformed center coordinates is then calculated. As all three photos are taken from the same bot's camera rotating along its pole, the circle center (the green dot in Figure 7a) marks the exact location of the bot.

5.3.3 App interface generation. The smartphone interface is automatically generated on the server and updated on the user's smartphone. In particular, the interface is generated as an HTML page based on the touchscreen interface being identified. The HTML page renders a list of textboxes and buttons based on the pre-labeled interface information in the server's database and is sent via Web-Socket. Once the bot activates a virtual button on the touchscreen of a kiosk and its interface changes, a new HTML page will be generated on the server accordingly and sent to the user's phone.

6 TECHNICAL EVALUATION

Toucha11y is essentially an open-loop system, meaning that whether it can help blind users successfully activate a touchscreen device depends on its own accuracy of localization, pole rotation, and reel extrusion. Thus, to understand the performance of our current working prototype, we conducted three technical evaluations.

6.1 Location Accuracy

The goal of this evaluation is to determine whether the Toucha11y bot can locate itself on touchscreens of various sizes and configurations. We collected five public kiosk interfaces, as shown in Figure 8. We began with four kiosk interface examples, each with a screen size of 12 inches, 21 inches, 27 inches, and 40 inches, representing a variety of use cases, including public lockers, airport kiosks, restaurant menus, and shopping mall navigation kiosks. As our algorithm is based on the number of visual features that can be detected on an image, we included a fifth interface example, a 12-inch touchscreen interface with a simple visual design and a monochrome background, to understand the prototype limitation.

6.1.1 Procedure. We located five testing points for each of the five touchscreen interfaces, four of which were located near the screen's four corners and one at the center. We assumed that blind users would find it easier to locate a kiosk device's physical bevel; thus,

Table 1: The distance between the calculated location and the actual location (unit: mm)

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Point 1	3.56	4.88	7.54	6.21	N/A
Point 2	6.08	N/A	4.06	8.72	N/A
Point 3	4.66	10.39	5.21	7.01	N/A
Point 4	4.11	11.98	6.12	10.02	N/A
Point 5	6.15	3.92	3.23	8.32	N/A
Mean	4.91	7.79	5.23	8.06	N/A

the corners might serve as physical references for blind users when placing Toucha11y devices. Similarly, the center of a screen might provide a sufficient flat surface area for blind users to work with.

For each testing point, we first created a red circular mark at the desired screen location. We then placed the Toucha11y bot right above the red mark. The bot would then proceed with the localization procedure as outlined in section 5.1.2, i.e., taking three consecutive photos of the screen with a 30° interval in between. The three photos were used to estimate the bot's location, as shown in Figure 9. Each testing location was evaluated three times, resulting in 15 data points for each screen interface example. We computed the error in the distance between the estimated and the actual coordinates for each data point.

6.1.2 Result. Table 1 summarizes the distance error between the calculated location and the actual location. Overall, the Toucha11y bot can be used to locate itself on touchscreen interfaces with different sizes and contents by taking a minimum of three photos with its camera. With our sample interfaces, the Toucha11y calculated its location with an average error distance of 6.50 mm (*SD* =
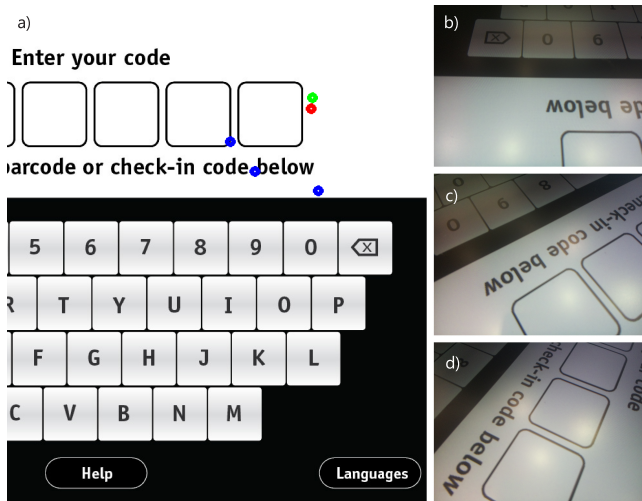

Figure 9: a) The red dot represents the actual location of the bot on the display, whereas the green dot represents the estimated location computed from the three blue dots based on the photos taken by the onboard camera, shown as b), c), and d).
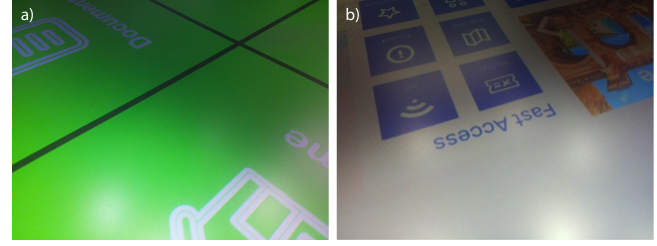

Figure 10: Toucha11y could not localize itself when the captured image has insufficient visual features.

1.66) for all but one point of the first four interfaces. The error distance is sufficient for many touchscreen interfaces, given that most touchscreen interface UIs will have buttons of the size of 25 mm by 25 mm [21]. For the failed case, i.e., the upper right corner of sample 2, as shown in Figure 10b, the bot was unable to compute its location due to the inefficient interface features being captured. This is identical to the situation with sample interface 5. When the touchscreen interface has fewer features across the screen, our current implementation is incapable of calculating its location.

Because the localization is primarily based on the implemented algorithm, we anticipate that the accuracy can be improved with software updates. For example, the current location of the bot is based on the center of a circle calculated from three photos taken with the bot, which is prone to error if one of the photos has an incorrect matching location. By using additional camera photos for the circle center calculation, we can potentially reduce this error and increase the precision of localization. Additionally, we can experiment with different image matching algorithms (e.g., [9, 31]), which may improve the system robustness under various touchscreen interface renderings (e.g. Figure 10).

6.2 Rotational Accuracy

To determine the rotational accuracy of our prototype, we commanded the Toucha11y bot to rotate to various degrees and then
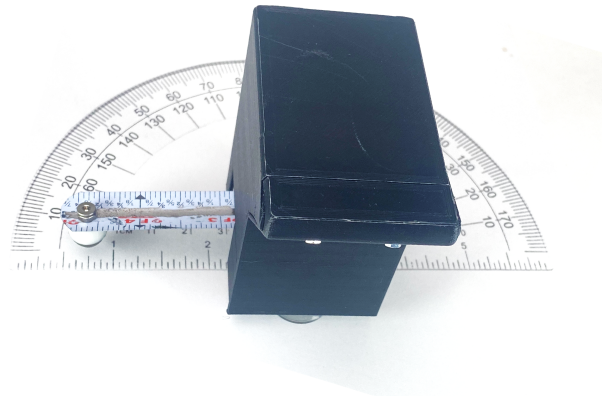

Figure 11: Rotational accuracy test with a protractor.

Table 2: The angular difference between the actual angle and the target angle (unit: degree)

	0	30	60	90	120	150	180
Clockwise	0	0.17	0.53	0.07	-0.30	-0.30	-0.60
Counter Clockwise	-0.67	0.23	-0.57	-0.07	0.10	0.40	0

measured the rotational error (Figure 11). In particular, the bot was first positioned in the center of a protractor aimed at angle 0°. It then rotated to 180° in 30° increments, and back. We repeated the experiment three times for each angle and reported the average of the three experimental results, keeping two decimal places.

6.2.1 Result. Table 2 reports the angular differences between the actual angle and the target angle in degrees. The average rotation error is 0.66 degrees (*SD* = 0.60). The error is primarily caused by the small backlash from 3D-printed gears and human manufacturing errors. The error can be further reduced with enhanced manufacturing and assembly processes.

6.3 Extension Accuracy

Finally, we evaluated the extension accuracy of the extendable reel. We instructed the bot to extend and retract in 50 mm steps between 0 and 700 mm. We ran three trials for each length and measured the actual length of extrusion.

6.3.1 Result. The average error for the extension length is 3.052 mm (*SD* = 3.147) in both extension and retraction actions. The linear encoder pattern at the bottom of the reel is the main cause of the error. As we affixed the reel with reflective tape at 2.5 mm spacing, the extension accuracy will not be greater than +-2.5 mm. Reducing the spacing will improve the extension accuracy.

7 USER STUDY

We conducted a formative user study to evaluate how Toucha11y supports blind users in using an inaccessible touchscreen interface.

7.1 Participants and Apparatus

We recruited seven participants (six female and one male) through online postings. Five participants self-reported as blind; two were low vision. All participants were familiar with accessibility features such as VoiceOver or TalkBack.

The study apparatus included the Toucha11y bot prototype, an Android- or iOS-based smartphone depending on the preference of the participant, and a 24-inch touchscreen display (Dell P2418HT [14]). To simulate the use of a real touchscreen kiosk, we prototyped an interactive bubble tea menu using Adobe XD [3].

7.2 Procedure

At the beginning of the user study, we collected the basic demographic information of the participants and then introduced the Toucha11y concept. Participants could ask any questions about the system and were encouraged to interact with the bot device, such

Figure 12: One participant is using a smartphone to explore the kiosk's touchscreen interface. The study was conducted at the participant's home.

as by touching it with their hands, putting it on the touchscreen display, or testing the smartphone app. After participants felt comfortable and were familiar with Toucha11y, they were given the study task, which was to order an "avocado tea with 50% sugar level" on an inaccessible touchscreen kiosk setup using the Toucha11y prototype. Specifically, participants would need to 1) place the Toucha11y bot on the touchscreen, 2) find the correct tea options from their smartphone with in-system accessibility apps (e.g., VoiceOver), 3) choose the sugar level, and 4) confirm the order details and complete the transaction. Figure 13 shows the Toucha11y bot's activating the corresponding buttons on the touchscreen setup. To ensure consistency, all participants were asked to follow the same ordering routine. The duration of task completion was recorded.

Following the study, we first collected Likert scale ratings. We then concluded with a semi-structured interview in which we solicited participants' comments and suggestions on the Toucha11y
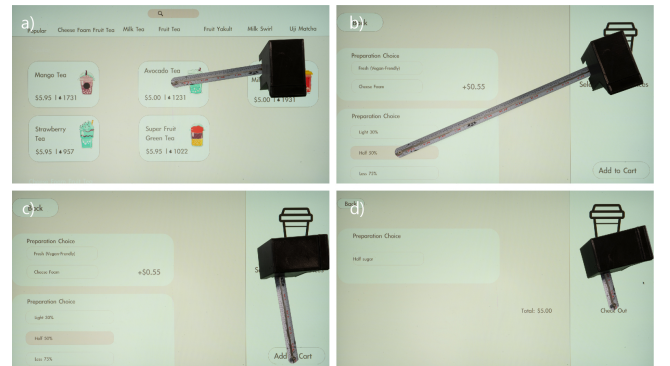

Figure 13: Study procedure. The blind users should make the purchase on their phone, with the Toucha11y bot completing the following actions: a) activate the "Avacado Tea" button, b) choose the half sugar option, c) triggers the "Add Cart" button, and d) "Check Out".

Table 3: Participants demographics.

ID	Gender	Vision Level	Hearing Level	Touchscreen Accessible Aids Software	Accessible Aids Software Use	Education Background
P1	Female	Blind	Good	Voice-over on iPhone	11 Years	Bachelor
P2	Female	Blind	Good	Voice-over on iPhone	12 Years	Master
P3	Male	Low vision	Good	Voice-over on iPhone	7 Years	Bachelor in progress
P4	Female	Blind	Good	Voice-over on iPhone	8 Years	Master
P5	Female	Blind	Good	Voice-over on iPhone, Braille display	4 Years	Master
P6	Female	Low vision	Good	Voice-over on iPhone, Braille display	Not sure, but proficient	Master
P7	Female	Blind	Good	Voice-over on iPhone	9 Years	Two-year college
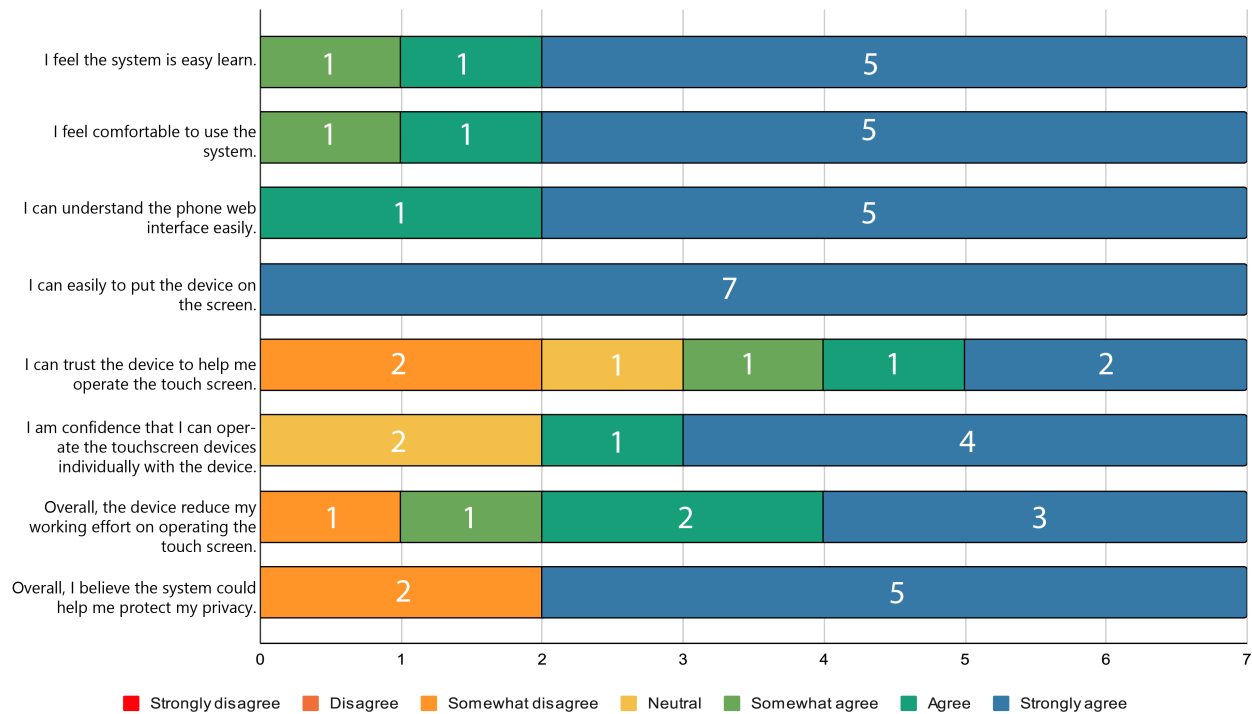
Figure 14: Self-reported ratings of using Toucha11y.

concept and the bot design. The research took about 40 minutes. Participants were paid at a rate of 30 USD per hour. For further analysis, the entire study was video- and audio-recorded.

7.3 Results

We present our user study result in this section and summarize the participants' feedback. Note that the Likert scale questions are ranged from 1 to 7; 1 refers to strongly disagree, and 7 refers to strongly agree.

All participants were able to complete the tea ordering task successfully. The average time of completion was 87.2 seconds (SD = 24.6). All of the participants found it easy to order with the smartphones, and most of the time was spent waiting for the bot to finish

the button activation events. We observed that five of the participants simply placed the Toucha11y bot close to the touchscreen kiosk's edge or a corner. Two participants placed the Toucha11y bot in the central location. As their placements happened to block the touchscreen buttons that needed to be triggered, participants were prompted to relocate the bot to a new location to restart the task. The relocation time was not factored into the completion time calculation.

As shown in the summarized self-reported rating (Figure 14), participants found that the Toucha11y device was easy to learn (M = 6.57, SD = 0.787), comfortable to use (M = 6.57, SD = 0.787), and very easy to put on the touchscreen (M = 7.00, SD = 0). The software interface provided sufficient instructions (M = 6.71, SD = 0.788).

They were confident that with Toucha11y they could operate a public touchscreen kiosk independently (M = 6.00, SD = 1.41).

Participants believed that the device could reduce their cognitive effort (M = 5.86, SD = 1.46) and protect their privacy (M = 5.86, SD = 1.95) while enabling them to operate touchscreen-based devices independently (M = 5.00, SD = 1.73). The concerns of those participants who rated lower on these questions were mainly related to their lack of familiarity with the device and that the study was not in a real situation. As P6 said, *"I am not sure I can trust this robot when I am in the grocery store alone, or if it can protect my privacy because you will not be there, and you can't teach me how to use this in the store... If I can practice more and be more familiar with the bot I could say yes...but right now I would say netural."*

Following the study, we solicited additional feedback on the Toucha11y concept from participants. All participants believed that the Toucha11y prototype could facilitate their use of touchscreen-based devices in public spaces. P7 stated, *"I could bring that device to the Social Security Office and put it on the screen. It would take pictures of the information and transfer it to my phone."* P1 also envisioned a variety of situations that Toucha11y to be useful. *"It can definitely help me at the doctor's office, airport, also DMV and other places. For my health insurance, I can use this on the company's kiosk, as it can help me check in and give me information, so I don't need to pay a person at the counter to get the information. Also touchscreen can be equally accessible to everyone."*

We also asked participants to provide feedback on the physical design of the Toucha11y bot. P2, P4, P5, and P6 stated that the device was easy to hold, had a good size, and would fit in their handbags or backpacks. However, for those who do not own a bag, the current prototype may be too large. P1 pointed out that the device was too large to fit in their bag or pocket, and if a user did not have a bag, the device was unlikely to be carried. P3 also expressed a desire for the size to be reduced so that they could be *"put in the pocket and carry it out"* like a phone.

Finally, the Toucha11y prototype also sparked discussions about the responsibility of technological accessibility. For example, P5 pointed out that the success of Toucha11y might have *"unintended consequences"*—that technology companies who build touchscreen kiosks *"would care even less (about accessibility) if they see this robot is so cool and using the phone and your app is not difficult at all."* *"Why do companies need to make technologies accessible?"*

8 DISCUSSION AND FUTURE WORK

8.1 Improve Toucha11y Design

8.1.1 Reduce the size. As a working prototype, the Toucha11y bot measures 50 mm by 70 mm by 93 mm. As P1 and P3 pointed out, the current prototype can be too big to be carried around as a personal device. Its size also precludes its use on smaller touchscreen interfaces, like those found on treadmills. We are considering several optimizations for the next iteration of the Toucha11y bot. For example, the Pi Zero W, which measures 65 mm by 30 mm, is the single largest component in the current bot design. A VoCore 2.0 controller [51] with roughly one-third the size (25.6 mm^2) but similar performance can be used as an alternative.
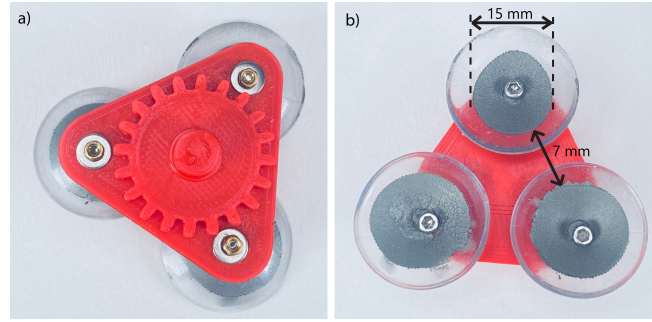

Figure 15: Suction cups with a conductive coating inside to trigger the touchscreen area below.

8.1.2 Cover touchable areas without relocation. As the Toucha11y bot has to be placed directly on top of the touchscreen surface, the bot base will inevitably cover a portion of the screen area, which sometimes can be the place that needs to be activated. In our current implementation, blind users must move the robot to a different location on the screen before proceeding. However, this solution may introduce additional errors during relocation.

To reduce the need for relocation, we are currently experimenting with a new bot base design in which the suction cups can directly trigger touch events. As shown in Figure 15, the inner surface of the base's suction cups (15 mm diameter) are coated with conductive paint, with a 7 mm gap in-between. Since the gap distance is less than the recommended side length of a touchscreen button [21], if the base is placed above an area to be activated, at least one suction cup will partially cover the button underneath and trigger accordingly.

We plan to evaluate the reliability of the new base design and eventually incorporate it into a future version of the Toucha11y system.

8.1.3 Improve privacy. Using Toucha11y can partially alleviate privacy concerns, as blind users can enter sensitive information directly from their smartphone rather than seek help from strangers. However, it is still possible for those standing behind the user to see the bot's actions as it registers touch events. A potential solution is to redesign the bot's camera so that, in addition to taking photos of the kiosk touchscreen, it can also scan the user's surroundings to detect overlooking. It is also possible to add a voice prompt to the smartphone interface to alert blind users.

8.1.4 Avoid pre-labeled interfaces in the database. Toucha11y assumes that the user interface of a touchscreen kiosk is pre-labeled and stored in a cloud-based database. The bot's camera is only to detect its placement, not to recognize the interface content. One limitation of this setup is that the bot cannot be used on a kiosk if its interface has not been pre-labeled or updated in the database.

One future direction would be to have the Toucha11y bot's built-in camera capture the entire screen area and leverage potential modern computer vision algorithms (such as Tesseract 4 [1]) for screen interface recognition. For this to work, the camera height must be adjustable, i.e., it must be able to rise above the touchscreen high enough to capture the full screen. While possible, enabling

the height-changing camera will likely increase the bot's size and complexity and thus require further investigation.

8.2 Long-Term Deployment

Although the user evaluation results were promising, Toucha11y was only tested in one simulated environment for a limited amount of time. We intend to deploy the device for long-term use in order to better understand how blind users will use it in real life. For long-term deployment, in addition to the hardware improvements already mentioned, we also need to update the system software. For example, we will need to gather a much larger database of the labeled touchscreen interface to cover a wide range of actual in-use touchscreen user interfaces. We anticipate that similar crowdsourcing approaches [18, 19] can be integrated with Toucha11y to enlarge the interface database.

8.3 Who is Responsible for Accessible Technology

Toucha11y is a temporary solution to make existing, inaccessible touchscreen devices accessible to blind users. We strongly concur with P5's remark, as we do not want the research on Toucha11y to be the excuse for technology companies to stop developing accessible kiosks. Instead, we wish more technology businesses would begin updating public kiosks with accessibility features (e.g., [13]) while also contributing the software interfaces of those inaccessible to the community so that they can be verified, labeled, and used by Toucha11y as a stopgap accessible solution.

8.4 Beyond Touchscreen Accessibility

Finally, while Toucha11y is designed to improve the accessibility of public kiosks for blind users, the notion of a small, personal device capable of mechanically interacting with the physical environment may present new opportunities for accessibility research. Looking forward, a Toucha11y-like bot could potentially be used to assist any people who have difficulty interacting with touchscreens, for example, people with motor disabilities. In the event that some touchscreen kiosks are poorly positioned and out of users' reach, a small mechanical bot with a long extendable reel can assist users in completing the touch interaction. If the bot's end-effector can be modified further, it could also be used to interact with physical gadgets beyond digital displays. For example, a Toucha11y-like bot with a universal gripper [6] may serve as a tabletop assistant. After scanning the desk, it can physically grab objects of interest, e.g., a pill bottle, for blind users. We note, however, that each of the potential opportunities calls for extra examination and deep involvement from all target users.

9 CONCLUSION

We have presented Toucha11y, a technical solution to enable blind users to use existing inaccessible touchscreen kiosks independently and with minimal effort. Toucha11y consists of a mechanical bot, a mobile interface, and a back-end server. The bot can be instrumented to an arbitrary touchscreen of a kiosk by the blind user to recognize its content, retrieve the corresponding information from a database, and render it on the user's smartphone. Through the smartphone app, a blind user can access the touchscreen content, and make selections using the built-in accessibility features of a smartphone. The bot can detect and activate the corresponding virtual button on the touchscreen. We presented the system design and a series of technical evaluations of Toucha11y. Through user evaluations, we concluded that Toucha11y could help blind users operate inaccessible touchscreen kiosks.

ACKNOWLEDGMENTS

REFERENCES

[1] 2022. *Tesseract documentation*. https://tesseract-ocr.github.io/ Accessed on 12.10.2022.
[2] ADA. [n.d.]. *Information and Technical Assistance on the Americans with Disabilities Act*. https://www.ada.gov/regs2010/2010ADAStandards/2010ADAstandards.htm Accessed on 09.06.2022.
[3] Adobe. 2022. *Adobe XD*. https://www.adobe.com/products/xd.html Accessed on 09.14.2021.
[4] Aipoly. [n.d.]. *Vision AI for the Blind and Visually Impaired*. https://www.aipoly.com/ Accessed on 09.06.2022.
[5] Majed Al Zayer, Sam Tregillus, Jiwan Bhandari, Dave Feil-Seifer, and Eelke Folmer. 2016. Exploring the use of a drone to guide blind runners. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. 263–264.
[6] John R Amend, Eric Brown, Nicholas Rodenberg, Heinrich M Jaeger, and Hod Lipson. 2012. A positive pressure universal gripper based on the jamming of granular material. *IEEE transactions on robotics* 28, 2 (2012), 341–350.
[7] Apple. [n.d.]. *Apple Accessibility*. https://www.apple.com/accessibility/vision/ Accessed on 09.06.2022.
[8] Mauro Avila Soto, Markus Funk, Matthias Hoppe, Robin Boldt, Katrin Wolf, and Niels Henze. 2017. Dronenavigator: Using leashed and free-floating quadcopters to navigate visually impaired travelers. In *Proceedings of the 19th international acm sigaccess conference on computers and accessibility*. 300–304.
[9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. Surf: Speeded up robust features. In *European conference on computer vision*. Springer, 404–417.
[10] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. 2010. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 333–342.
[11] Dimitrios Dakopoulos and Nikolaos G Bourbakis. 2009. Wearable obstacle avoidance electronic travel aids for blind: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, 1 (2009), 25–35.
[12] Josh Urban Davis, Te-Yen Wu, Bo Shi, Hanyi Lu, Athina Panotopoulou, Emily Whiting, and Xing-Dong Yang. 2020. TangibleCircuits: An interactive 3D printed circuit education tool for people with visual impairments. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
[13] Katie Deighton. [n.d.]. *McDonald's Starts Making Self-Service Easier for Blind Diners*. https://www.wsj.com/articles/mcdonalds-starts-making-self-service-easier-for-blind-diners-11631900280 Accessed on 09.06.2022.
[14] Dell. 2022. *Dell 24 Touch Monitor - P2418HT*. https://www.dell.com/en-us/shop/dell-24-touch-monitor-p2418ht/apd/210-alcs/monitors-monitor-accessories Accessed on 09.14.2021.
[15] Wafa Elmannai and Khaled Elleithy. 2017. Sensor-based assistive devices for visually-impaired people: Current status, challenges, and future directions. *Sensors* 17, 3 (2017), 565.
[16] Google. [n.d.]. *Android Accessibility Overview*. https://support.google.com/accessibility/android/answer/6006564?hl=en Accessed on 09.06.2022.
[17] 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* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 651–664. https://doi.org/10.1145/2984511.2984518

[18] Anhong Guo, Jeeeun Kim, Xiang'Anthony' Chen, Tom Yeh, Scott E Hudson, Jennifer Mankoff, and Jeffrey P Bigham. 2017. Facade: Auto-generating tactile interfaces to appliances. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 5826–5838.

[19] Anhong Guo, Junhan Kong, Michael Rivera, Frank F Xu, and Jeffrey P Bigham. 2019. Statelens: A reverse engineering solution for making existing dynamic touchscreens accessible. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 371–385.

[20] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. 2020. Captioning images taken by people who are blind. In *European Conference on Computer Vision*. Springer, 417–434.

[21] Aurora Harley. 2019. *Touch Targets on Touchscreens*. https://www.nngroup.com/articles/touch-target-size/ Accessed on 09.14.2021.

[22] 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*. 73–80.

[23] 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*. 347–350.

[24] Shaun K Kane, Chandrika Jayant, Jacob O Wobbrock, and Richard E Ladner. 2009. Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*. 115–122.

[25] Shaun K Kane, Meredith Ringel Morris, and Jacob O Wobbrock. 2013. Touchplates: low-cost tactile overlays for visually impaired touch screen users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–8.

[26] Varsha Koushik, Darren Guinness, and Shaun K Kane. 2019. Storyblocks: A tangible programming game to create accessible audio stories. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[27] Vladimir Kulyukin, Chaitanya Gharpure, and John Nicholson. 2005. Robocart: Toward robot-assisted navigation of grocery stores by the visually impaired. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2845–2850.

[28] Vladimir Kulyukin, Chaitanya Gharpure, John Nicholson, and Grayson Osborne. 2006. Robot-assisted wayfinding for the visually impaired in structured indoor environments. *Autonomous robots* 21, 1 (2006), 29–41.

[29] Vladimir Kulyukin, Chaitanya Gharpure, John Nicholson, and Sachin Pavithran. 2004. RFID in robot-assisted indoor navigation for the visually impaired. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, Vol. 2. IEEE, 1979–1984.

[30] Jonathan Lazar, J Bern Jordan, and Gregg Vanderheiden. 2019. Toward unified guidelines for kiosk accessibility. *Interactions* 26, 4 (2019), 74–77.

[31] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. 2011. BRISK: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*. Ieee, 2548–2555.

[32] Jiasheng Li, Zeyu Yan, Ebrima Haddy Jarjue, Ashrith Shetty, and Huaishu Peng. 2022. TangibleGrid: Tangible Web Layout Design for Blind Users. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 47, 12 pages. https://doi.org/10.1145/3526113.3545627

[33] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.

[34] Natalina Martiniello, Werner Eisenbarth, Christine Lehane, Aaron Johnson, and Walter Wittich. 2022. Exploring the use of smartphones and tablets among people with visual impairments: Are mainstream devices replacing the use of traditional visual aids? *Assistive Technology* 34, 1 (2022), 34–45.

[35] Microsoft. [n.d.]. *Seeing AI*. https://www.microsoft.com/en-us/ai/seeing-ai Accessed on 09.06.2022.

[36] Rezylle Milallos, Vinita Tibdewal, Yiwen Wang, Andre Ogueh Udegbe, and Tae Oh. 2021. "Would the smart cane benefit me?": Perceptions of the Visually Impaired towards Smart Canes. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–3.

[37] Rezylle Milallos, Vinita Tibdewal, Yiwen Wang, Andre Udegbe, and Tae Oh. 2022. An Exploratory Study on the Low Adoption Rate of Smart Canes. In *International Conference on Human-Computer Interaction*. Springer, 492–508.

[38] Laura Miller. [n.d.]. *The Legal Case: Why Your Kiosk Should be Accessible*. https://www.tpgi.com/the-legal-case-why-your-kiosk-should-be-accessible/ Accessed on 09.06.2022.

[39] Tim Morris, Paul Blenkhorn, Luke Crossey, Quang Ngo, Martin Ross, David Werner, and Christina Wong. 2006. Clearspeech: A Display Reader for the Visually Handicapped. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14, 4 (2006), 492–500. https://doi.org/10.1109/TNSRE.2006.881538

[40] Marius Muja and David G Lowe. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)* 2, 331-340 (2009), 2.

[41] 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*. 13–20.

[42] John Nicholson, Vladimir Kulyukin, and Daniel Coster. 2009. ShopTalk: independent blind shopping through verbal route directions and barcode scans. *The Open Rehabilitation Journal* 2, 1 (2009).

[43] KNFB Reader. [n.d.]. *OneStep Reader Website*. https://knfbreader.com/ Accessed on 09.06.2022.

[44] Armin Ronacher. 2010. *Flask documentation*. https://flask.palletsprojects.com/en/2.0.x/ Accessed on 08.15.2021.

[45] Lee Stearns, Ruofei Du, Uran Oh, Catherine Jou, Leah Findlater, David A Ross, and Jon E Froehlich. 2016. Evaluating haptic and auditory directional guidance to assist blind people in reading printed text using finger-mounted cameras. *ACM Transactions on Accessible Computing (TACCESS)* 9, 1 (2016), 1–38.

[46] Ryo Suzuki, Clement Zheng, Yasuaki Kakehi, Tom Yeh, Ellen Yi-Luen Do, Mark D Gross, and Daniel Leithinger. 2019. Shapebots: Shape-changing swarm robots. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 493–505.

[47] Hussain Tinwala and I. Scott MacKenzie. 2010. Eyes-Free Text Entry with Error Correction on Touchscreen Mobile Devices. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (Reykjavik, Iceland) *(NordiCHI '10)*. Association for Computing Machinery, New York, NY, USA, 511–520. https://doi.org/10.1145/1868914.1868972

[48] Gregg C. Vanderheiden. [n.d.]. Flexible access system for touch screen devices. https://patents.google.com/patent/US6049328 US Patent 6,049,328A.

[49] Gregg C Vanderheiden. 1996. Use of audio-haptic interface techniques to allow nonvisual access to touchscreen appliances. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 40. SAGE Publications Sage CA: Los Angeles, CA, 1266–1266.

[50] Ramiro Velázquez. 2010. Wearable assistive devices for the blind. In *Wearable and autonomous biomedical devices and systems for smart environment*. Springer, 331–349.

[51] VoCore. [n.d.]. *Coin-sized Linux Computer With WiFi and Ethernet*. https://vocore.io/ Accessed on 08.15.2022.

[52] W3C. 2018. *Web Content Accessibility Guidelines (WCAG) 2.1*. https://www.w3.org/TR/WCAG21/ Accessed on 08.15.2022.

[53] Mohd Helmy Abd Wahab, Amirul A Talib, Herdawatie A Kadir, Ayob Johari, Ahmad Noraziah, Roslina M Sidek, and Ariffin A Mutalib. 2011. Smart cane: Assistive cane for visually-impaired people. *arXiv preprint arXiv:1110.5156* (2011).

[54] Hanlu Ye, Meethu Malu, Uran Oh, and Leah Findlater. 2014. Current and future mobile and wearable device use by people with visual impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3123–3132.

[55] Georgios Yfantidis and Grigori Evreinov. 2006. Adaptive blind interaction technique for touchscreens. *Universal Access in the Information Society* 4, 4 (2006), 328–337.

[56] Shengdong Zhao, Pierre Dragicevic, Mark Chignell, Ravin Balakrishnan, and Patrick Baudisch. 2007. Earpod: Eyes-Free Menu Selection Using Touch Input and Reactive Audio Feedback. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1395–1404. https://doi.org/10.1145/1240624.1240836

[57] Yuhang Zhao, Sarit Szpiro, and Shiri Azenkot. 2015. Foresee: A customizable head-mounted vision enhancement system for people with low vision. In *Proceedings of the 17th international ACM SIGACCESS conference on computers & accessibility*. 239–249.

[58] Sam Zietz. 2020. *Can Kiosks Help Restaurants Stay Alive Through COVID-19?* https://www.qsrmagazine.com/outside-insights/can-kiosks-help-restaurants-stay-alive-through-covid-19 Accessed on 08.15.2022.