Deploying VizLens: Characterizing User Needs, Preferences, and Challenges of Physical Interfaces Usage in the Wild

Andi Xu[*]
University of Michigan
Ann Arbor, MI, USA
andixu@umich.edu

Mahdi Qazwini[*]
University of Michigan
Ann Arbor, MI, USA
mqazwini@umich.edu

Chen Liang
University of Michigan
Ann Arbor, MI, USA
clumich@umich.edu

Anhong Guo
University of Michigan
Ann Arbor, MI, USA
anhong@umich.edu

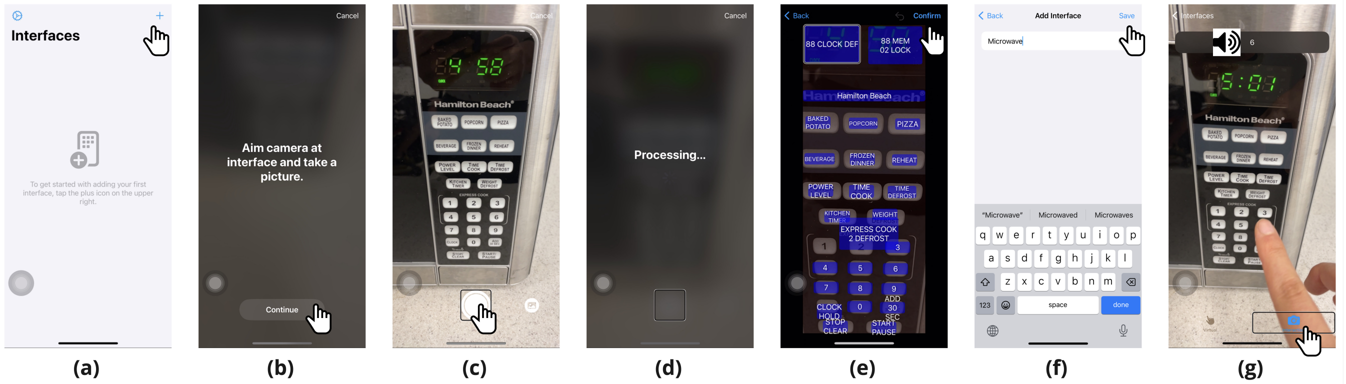

Figure 1: VizLens Main User Interface. A user can add an interface by taking a photo of the interface (a-c). After the system process the interface image (d), users can explore and edit detected buttons (e), name it (f) and use it in live interaction mode (g).

ABSTRACT

Blind or Visually Impaired (BVI) people often encounter flat, inaccessible interfaces. Current solutions lack cost-effectiveness, portability, and robustness in real-world settings. We introduce VizLens, a fully-automated, full-stack mobile application powered by computer vision algorithms. The system is deployed and publicly available through the Apple App Store (https://vizlens.org/). From May to August 2023, we had 665 users, who uploaded 1,320 interface images. We aim to use it to study usage patterns and possible challenges BVI users may encounter with flat interfaces through a large-scale study in real-world settings. With in-depth analysis of user data and activity logs, our study will provide insights into BVI users' interface interests, preferred assistance modes, and potential challenges due to system limitations or users' diverse abilities. Our goal is to enhance the understanding of how BVI users interact with inaccessible, flat interfaces, and inform future assistive technology design.

CCS CONCEPTS

• **Human-centered computing → Accessibility systems and tools**; **Empirical studies in HCI**.

KEYWORDS

physical interfaces, appliances, blind, visually impaired people, accessibility, computer vision, deployment

ACM Reference Format:
Andi Xu, Mahdi Qazwini, Chen Liang, and Anhong Guo. 2023. Deploying VizLens: Characterizing User Needs, Preferences, and Challenges of Physical Interfaces Usage in the Wild. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '23), October 22–25, 2023, New York, NY, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3597638.3614493

1 INTRODUCTION

Flat interfaces are commonly found in daily lives. From microwave control pads to kiosks at restaurants, interacting with these interfaces has become an essential step to perform tasks independently. However, many of them remain inaccessible to blind or visually impaired (BVI) users. The lack of audio or tactile feedback makes it challenging or impossible for BVI users to independently explore possible options on the interface and navigate toward the button they want to press. This may require assistance from sighted people nearby, which may not always be available.

Various commercial and research solutions have been proposed to tackle this problem. Recent products like Be My Eyes [4] and Aira [3] offer remote assistance via mobile devices, but have problems with availability or high cost. Prior studies have tried approaches to use computer vision [5, 10, 11], crowd-sourcing [7], or both [6, 8] to generate guidance. However, some of the approaches need additional cameras and still have problems in terms of cost and portability [11]. Robustness is another issue as they cannot cover a wide range of use cases in the real world [5] and fail to provide enough trustworthiness and independence for BVI people.

[*] Both authors contributed equally to this work.

ASSETS '23, October 22–25, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0220-4/23/10.
https://doi.org/10.1145/3597638.3614493

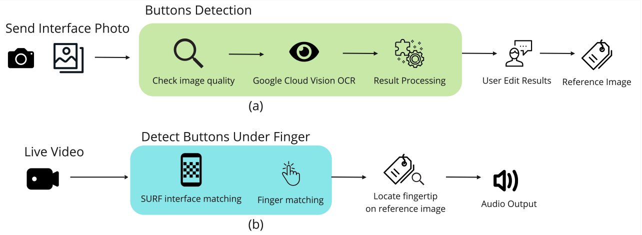

Figure 2: VizLens System Diagram. (a) Creating an interface: Users can send a photo of their interface and the system will process and generate a labeled reference image. (b) Using the Interface: the system locates the interface and fingertip from the video frame based on the reference image, and announces the button being pointed.

Therefore, we introduce VizLens, a full-stack mobile application upgraded from the previous prototype that enables users to access interfaces in daily life. The system is backed up by computer vision algorithms, thus fully automated and efficient. It is deployed and publicly available through the Apple iOS App Store (https://vizlens.org/), and we hope to use this system to study BVI people's usage patterns and possible challenges on flat interfaces through a large-scale study in real-world settings. With the permission from users, we collect user data and activity logs, aiming to understand i) what interfaces BVI people are interested in; ii) modalities they prefer when getting assistance; and iii) challenges they may encounter, which could arise from our system limitations or some specific tasks. From 05/01/2023 to 08/17/2023, we had 665 users, who uploaded 1,320 interface images. Overall, we aim to use this deployment study to better understand BVI people's interaction with flat, inaccessible interfaces and provide design implications for future assistive technologies.

2 VIZLENS SYSTEM

VizLens is an accessible system that can help BVI users use appliances with flat touch interfaces. It consists of an iOS app for user interaction and a Firebase database to sync user data.

2.1 Mobile App and User Interface

VizLens, powered by Firebase, supports user authentication and can sync users' data over different devices. Users will be able to add an image of an appliance either from the camera or album. The app will process the image and return a labeled interface image, which will be stored as a reference image. Details of image processing will be discussed in section 2.2. Then, the user can access the added interface from the main screen and start exploring the interfaces via either the virtual mode or live interaction mode. Virtual mode aims to help users learn the layout of their interface. Its modality is similar to using VoiceOver: the screen will show the labeled interface image, and audio will read out any text under their finger (Figure 1g). Users can also swipe left and right to explore buttons. If they believe the detected text has errors, users can modify the machine-detected labels in this virtual mode through the pop-out menu with a list of actions, including deleting, splitting (e.g., split button "express cook 2 defrost" into 4 buttons) (Figure 3a), and merging buttons (e.g., merge buttons "defrost", "weight" into one) (Figure 3b).

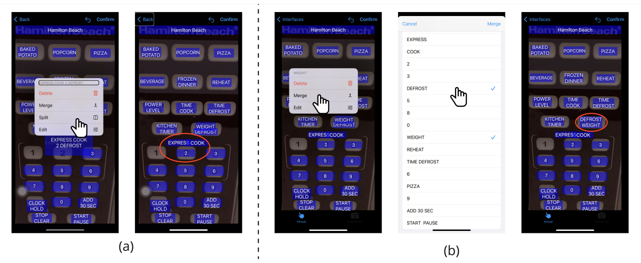

Figure 3: Edit Button Interface. Users can split a button (a) or merge multiple buttons (b) in the virtual model.

The live interaction mode helps users use an interface while moving their finger on the physical appliance interface Figure 1. After entering this mode, a message will show up to tell the user to aim their camera at the interface as steadily as possible and move their fingers on the interface. The app will first try to locate the interface using the reference image and announce audio feedback "Aim camera at appliance." When successfully locating the appliance, background music will play and users can start putting their finger onto the appliance. The system will locate the user's fingertip and match its position with the reference image, reading out any text being touched by the user. The details of interface and fingertip matching will be discussed in section 2.3.

2.2 Interface Image Processing

The image processing part labels the interface image. To ensure that the image has good quality so that models can perform well, the app first will examine the image quality in terms of blurriness and lighting by calculating a score for each to check whether it meets a pre-defined threshold. The blurriness score is calculated by converting the image's color space from RGB to grayscale, then calculating the variance of grayscale's Laplacian. The lighting value is acquired by getting the mean of the HSV color space of the image.

Secondly, we use Google Cloud Vision API [2] to perform OCR on the image to find text-labeled buttons. A limitation of this method is that icon-based buttons cannot be detected, which requires future work. Successful results will contain detected text and their boundaries in sentence, phrase, and word levels. Results with low level of confidence will be excluded. We generate approximate button boundaries for each word, merge buttons that overlap or are too close to each other, and expand the machine-returned borders proportionally based on the interface size and button numbers. Here, although some texts on the interfaces might not be buttons thus are false positives, users can easily remove them, which could be easier to modify than the cases of missing buttons. Additionally, although the button boundaries are estimated, we designed them to have small margins to lower the risk of users missing the button. Finally, the app will build a feedback look-up table for image grids and buttons.

2.3 Interface and Finger Matching in Live Interaction Mode

Another major yet challenging part of the VizLens system is detecting the text pointed (and covered) by users' fingers in the live mode. After some iterations and experimentation, to locate the interface from the camera and match it with the reference image in storage, we settled with Speeded Up Robust Features (SURF) [1], using FLANN with knn. To locate the fingertip, we first let users specify which finger they would use to point in the settings, then use Apple Vision hand recognition model [9] to find the point in the camera, and use SURF again to translate it into the position on the reference image.

2.3.1 Design and Technical Iterations. We tried to use a QR code for the interface matching issue: we would generate a QR code for each reference image and let the user place a printed code beside the interface so that later the system can use the code as an anchor. However, this approach requires additional user effort to print out the code and is not sufficiently lightweight.

As for fingertip matching, our previous attempt of using skin color thresholding [12] shows less robust results on fingertip detection. We observed that BVI users may navigate with the hand open instead of doing a pointing gesture when using appliances. With the pointing gestures, different users may prefer different fingers for exploration. All these motivate us to switch to hand tracking with Apple Vision, which enables the system to detect any finger's position and enables users to choose their preferred fingers.

2.3.2 Limitations. Despite the improvements in supporting diverse use cases, the SURF method currently cannot perform very well in the following two situations: i) appliances with shiny or reflective surfaces, which brings inconsistent visual features that are challenging to match; ii) Interface matching algorithms fail to find enough feature matches if a large portion of the interface is covered by the hand. However, the more parts of the hand is shown, the better the hand tracking algorithms can perform. The nature of the task causes a trade-off.

3 DATA COLLECTION AND FUTURE WORK

VizLens is published as a free app with the option for users to decide whether to participate in our IRB-approved research study at any time. While necessary data is securely stored on the Firebase, only when a user opts into the IRB we will use their data for our study. The research data is not stored with personal information like names and emails, but tagged by anonymous user IDs. In this section, we will introduce the user data collected by us and research questions we can answer given those data and future work.

3.1 Data Collection

Upon registration, the user is asked whether they would like to opt into the IRB with study details. For IRB opted-in users, we collect and analyze their app usage logs that primarily focus on two aspects: (1) Interfaces: including the interface image, detected buttons, frequency of using it on both virtual and interaction modes, timestamps, and low-quality images that were not able to be automatically processed. (2) App usage: a list of activity logs with timestamps, including session start and end time, sequence of views, interactions, errors, usage time when navigating interfaces, and device angles while being used by the user, specifically when the user is taking the photo or use the live interactive mode (which requires pointing the camera to the interface).

3.2 Research Questions

3.2.1 RQ1: What kind of interfaces are BVI users mostly interested to use? Based on the interfaces users added, we would like to analyze the distribution of interface types. Interfaces could be grouped in terms of in-home appliances versus public ones, buttons with text versus buttons with icon, physical interface (e.g. microwave) versus digital ones (tablet). We hope to answer those questions based on the interface data (both successful and failed ones) we collect from users by building an affinity diagram. Currently, most of the appliances are physical ones from home with text-based buttons, like microwaves and ovens.

3.2.2 RQ2: How do BVI users use VizLens, such as their preference of Virtual Mode vs. Live Interactive Mode? With activity logs, we are able to analyze and compare the time users spend in virtual mode versus live interaction mode. If a user uses virtual mode a lot, what could be the reason, and what information are they trying to get from it? Is it because they find interaction mode not very effective? Or they are able to use the appliance directly after learning the layout of all buttons on the screen? We also plan to interview or set up surveys with active users to learn from their experience in more detail.

3.2.3 RQ3: What are some challenges that happen during a user journey? When a user tries to use a flat interface with Vizlens, challenges could come from the system side, which means errors brought by system's performance like OCR detection accuracy and SURF matching accuracy, and from the user side, such as not able to aim camera at the interface well. Understanding those challenges could provide design implications for future assistive tools. We log all errors users would encounter during their usage of our app, including system-side errors (e.g., bugs, internet connection), and user-side unsuccessful attempts to use the app (e.g., failure to add images because of image quality). Among 317 users who encountered errors, 48 encountered interface uploading failure because of bad Internet connection.

Furthermore, 68 users encountered failure to take qualified images, and there are 654 failed attempts to upload images in total. From invalid image dataset and users' email feedback, we learned that it is hard for users to i) aim the camera steadily; ii) know their surrounding's lighting condition; iii) ensure important elements are captured by the camera; iv) know whether their appliance has text on buttons, or just icon, which cannot be detected by OCR. The camera aiming issue also persists in live interaction mode, which also requires users to aim the camera at the interface and the finger.

To address this, besides fine-tuning the image quality thresholds, we plan to use invalid images and Gyroscope information we collected to understand how BVI users hold their devices when taking pictures, to get insights on designing better camera aiming techniques and prompts.

ACKNOWLEDGMENTS

We thank our users for contributing to our deployment data collection and for providing feedback. This research is supported in part by a Google Cloud Platform Credit Award. We thank Yatharth Chhabra for his help with early stage technical iterations.

REFERENCES

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. SURF: Speeded Up Robust Features. In *Computer Vision – ECCV 2006*, Aleš Leonardis, Horst Bischof, and Axel Pinz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 404–417.

[2] Google Cloud. 2023. Google Cloud Vision. https://cloud.google.com/vision

[3] Aira Tech Corp. 2022. Aira. https://aira.io/

[4] Be My Eyes. 2023. Be My Eyes. https://www.bemyeyes.com/

[5] Giovanni Fusco, Ender Tekin, R.E. Ladner, and James Coughlan. 2014. Using Computer Vision to Access Appliance Displays. *ASSETS / Association for Computing Machinery. ACM Conference on Assistive Technologies* 2014. https://doi.org/10.1145/2661334.2661404

[6] 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

[7] 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* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 5826–5838. https://doi.org/10.1145/3025453.3025845

[8] 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* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 371–385. https://doi.org/10.1145/3332165.3347873

[9] Apple Machine Learning. 2023. Apple Machine Learning Vision. https://developer.apple.com/documentation/vision/

[10] Chen Liang, Yasha Iravantchi, Thomas Krolikowski, Ruijie Geng, Alanson P. Sample, and Anhong Guo. 2023. BrushLens: Hardware Interaction Proxies for Accessible Touchscreen Interface Actuation. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3586183.3606730

[11] 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

[12] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreeva. 2004. A Survey on Pixel-Based Skin Color Detection Techniques. (03 2004).