Access Lens: A Gesture-Based Screen Reader for Real-World Documents

Shaun K. Kane, Brian Frey
Department of Information Systems
UMBC
Baltimore, MD 21250 USA
{skane, frey1}@umbc.edu

Jacob O. Wobbrock
Information School | DUB Group
University of Washington
Seattle, WA 98195 USA
wobbrock@uw.edu

ABSTRACT
Gesture-based touch screen user interfaces, when designed to be accessible to blind users, can be an effective mode of interaction for those users. However, current accessible touch screen interaction techniques suffer from one serious limitation: they are only usable on devices that have been explicitly designed to support them. *Access Lens* is a new interaction method that uses computer vision-based gesture tracking to enable blind people to use accessible gestures on paper documents and other physical objects, such as product packages, device screens, and home appliances. This paper describes the development of Access Lens hardware and software, the iterative design of Access Lens in collaboration with blind computer users, and opportunities for future development.

Author Keywords: Accessibility; blindness; gestures; computer vision; augmented reality.

ACM Classification Keywords: H.5.2 [Information Interfaces and Presentation]: User Interfaces - Input devices and strategies.

INTRODUCTION
Until recently, many mainstream touch screen applications were inaccessible to blind people. However, in the past several years, a number of research projects (*e.g.,* [4]) have demonstrated that, by combining audio or tactile output with accessible gestures, blind people can effectively use touch screen interfaces, even if they are unable to see the screen. The creators of mainstream touch screen devices have incorporated some of these accessible gestures into their products, and many devices now provide accessible gestures for blind users.

Although touch screen accessibility has improved in recent years, many touch screen devices are still inaccessible. Furthermore, accessible gestures themselves suffer from a fundamental limitation: they are only usable on devices and applications that have been explicitly designed to support them. For example, a blind smartphone user may use accessible gestures to interact with her favorite apps, but will be unable to use those gestures to read her paper mail or read a campus map.

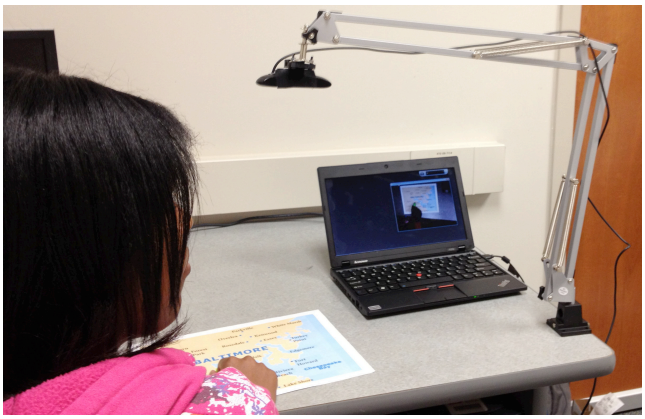

Figure 1. A blind person uses Access Lens to read a paper map using gestures. Access Lens recognizes the labels on the map and reads them as the user touches them with her finger.

To explore the potential of applying accessible gestures to the physical documents, we introduce Access Lens (Fig. 1), a system that allows blind users *to apply accessible gestures to real-world objects,* including paper documents. Access Lens (AL) uses a camera and computer vision to identify and recognize text in the environment, and tracks the user's hands in space, describing objects in the environment using synthesized speech. As a result, AL enables users to explore otherwise inaccessible objects using accessible gestures.

In this paper, we describe the design of Access Lens, including its computer vision and gesture tracking techniques. We also describe a formative study in which 5 blind computer users tested the prototype, and discuss opportunities for future development of the AL platform.

RELATED WORK
Reading Aids for Blind People
Prior projects such as the KNFB Reader Mobile[1] have used optical character recognition (OCR) to capture an image of

[1] http://www.knfbreader.com

CHI 2013, April 27–May 2, 2013, Paris, France.

a document and read it out using synthesized speech. However, AL provides the additional feature of exploring documents using gestures, which makes it ideal for exploring complex documents and spatial data. VizWiz [1] allows blind people to ask questions about their surroundings by taking a photograph and dictating a query, which is answered by a remote human worker. VizWiz provides limited support for searching for objects by moving the camera through space, but does not currently support gesture control, and relies upon remote human workers for feedback. AL uses automated recognition techniques, and can be used offline and in other situations in which the user does not wish to involve human workers.

Augmented Workspaces

AL was inspired by prior augmented workspaces such as the DigitalDesk [11] and Bonfire [3]. These projects used computer vision to track a user's hand gestures over a surface, enabling them to interact with virtual information in an otherwise uninstrumented environment. However, these systems focused primarily on visual interaction for sighted users. In contrast, AL is designed to enable blind users to explore objects using gestures. EyeRing [8] is a finger mounted camera that can be used to provide blind people with information about text or objects in their environment. AL has a similar goal, but uses a very different form factor (a mounted camera *vs.* a finger-worn camera), and leverages this form factor to support gestural exploration and spatial guidance.

DESIGN OF ACCESS LENS

The AL system comprises: (1) camera hardware mounted in the user's workspace; (2) computer vision software for recognizing text and tracking gestures. AL was developed through iterative testing with blind computer users.

Hardware

The AL prototype comprises a Logitech C910 webcam mounted on a lamp arm, connected to an Apple MacBook Air laptop with a dual-core 2.13 GHz processor and 4GB RAM (Fig. 1). However, AL's camera could be attached to any desktop or laptop PC. One benefit of a desk-mounted configuration is that the user does not need to aim the camera, which can be difficult for blind people [12].

Software

AL's primary functions are: scanning text in documents and other objects, and enabling the user to explore text using hand gestures and speech commands. AL was implemented using Python 2.7 on a laptop using Windows 7. AL uses OpenCV [2] for core computer vision algorithms, and Microsoft's .NET speech libraries for spoken commands.

When launched, AL captures an image of the user's workspace, and uses background subtraction [9] to identify new objects placed in the workspace. When the user wishes to scan an object, he or she places the object within the workspace and presses a predefined *Scan* key on the computer keyboard. AL locates the largest foreground object and attempts to scan its text. As AL is primarily intended to scan paper documents, and because the camera may be placed at an oblique angle relative to the scanned object, AL attempts to identify the corners of a rectangular object and de-skew that object using a homographic transform [10]. The user is notified if a rectangle cannot be found, such as when a document is partially outside the camera's view, or if a non-rectangular object is scanned.

When an object has been detected, AL identifies likely text regions [13] and passes them to an OCR engine.[2] Recognized text may be optionally checked against a dictionary file, and corrected by finding a match with the minimum string distance [6]. Once the text has been recognized, the user's hand is identified by combining the foreground/background model with a color-based skin detector [7]. AL assumes that the user is pointing with a single finger (as in Fig. 2), and identifies the fingertip as the point on the hand furthest from the user's body location.

Interaction Modes

AL provides several methods for reading document text using hand gestures and speech commands. In its most basic mode, known as *direct touch mode*, AL tracks the user's fingertip and speaks the text closest to it (Fig. 2). Direct touch mode enables blind users to read previously inaccessible documents simply by touching them. Furthermore, because text is read as the user touches it, the user may also learn the document's spatial layout.

While direct touch mode provides the ability to read previously inaccessible documents, exploring a document using direct touch and audio can be difficult, especially when trying to locate a specific item. To improve browsing efficiency, AL offers two supplementary navigation features: *edge menus* and *voice commands.*

Edge menus: To enable faster browsing of scanned objects, AL provides a virtual *edge menu overlay* inspired by Access Overlays [5]. When edge menus are activated, AL adds a column of virtual buttons along the right edge of the current object (*e.g.,* at the right edge of the scanned page). This menu lists all recognized text fragments in alphabetical order. Touching a button on the edge reads the name of that item. Dwelling on a menu item causes AL to provide *guided speech directions* to the item's actual location (*e.g.,* "Down, down, left, …"), as in Access Overlays [5]. Figure 2 shows an edge menu along the right edge of the page.

Voice commands: AL's voice commands are activated by pressing a key on the laptop keyboard. Saying *"list"* causes AL to read all on-screen items. Saying *"find <item>"* provides guided speech directions to the named item. Voice commands may also be used to scan a new document or to toggle edge menus.

[2] ABBYY FineReader (www.abbyy.com)

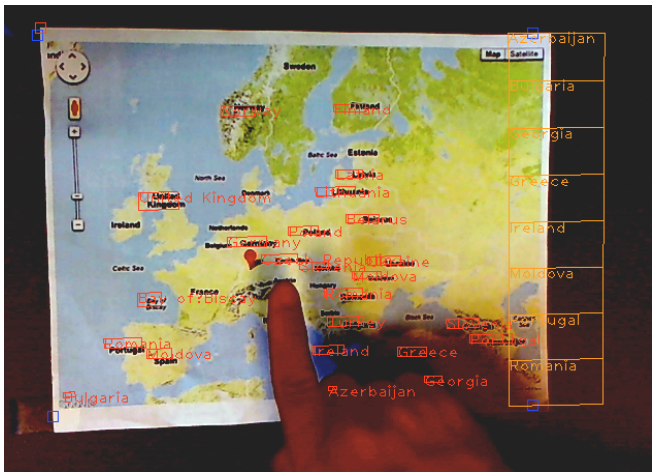

Figure 2. Access Lens viewing a map. Touching near the map labels speaks their names. An *edge menu* along the right edge of the page shows a sorted index of on-screen items.

While AL is focused on reading text, it also supports a *color identification mode*. When this mode is active, AL names the color closest to the user's finger. This mode can be used to scan documents, clothing, and other objects.

FORMATIVE EVALUATION OF ACCESS LENS

As AL offers a fundamentally new user interface for blind people, we were initially unsure whether users would even be able to use it. We tested early prototypes of AL with 5 blind computer users (3 female, average age 34.8). Data from these sessions was used to refine parameters for the underlying algorithms. Pilot participants were enthusiastic about AL, and suggested new uses for the system, such as color identification, which we added to the final prototype.

We then conducted a formative evaluation of AL with 5 blind users (3 female, average age 33.8). All participants were regular computer users; 2 used a screen reader exclusively, while 3 used both a screen reader and a screen magnifier. All participants had previously used a touch screen-based device, and 4 owned such a device. Three participants had participated in the pilot tests of AL. The goals of the evaluation were: 1) to collect participants' feedback about the AL; 2) to compare AL's interaction modes; 3) to compare AL across document types; and 4) to identify potential usability and reliability challenges.

Each participant used AL for one 60-minute session. Each participant received a 10-minute introduction to AL and each method of interaction (direct touch, edge menus, voice commands, and color identification), using a US state map. Participants then used AL to freely explore 3 documents for approximately 10 minutes each: a *diagram* of the human body, a *map* of Europe (Fig. 2), and a *table* containing political poll results. Participants tested *color identification mode* for approximately 5 minutes using these documents and an image of a US state flag. A member of the research team was present throughout the activity to place the documents in the workspace, to observe the participant, and to answer any questions about the task or about the document being explored. Following the session, participants provided verbal feedback about their experiences, and rated each interaction method and document type on a 7-point Likert scale (Table 1).

Observations from the Formative Evaluation

Overall usefulness: Overall, participants were very enthusiastic about AL. When asked to rate the overall usefulness of AL on a 7-point Likert scale, participants gave AL a median rating of 6, and all but one rated it 6 or higher. Participants expressed interest in using AL to read maps, charts, bus schedules, bills, sheet music, magazines, medical documents, and clothing.

Interaction modes: Participants provided ratings for each interaction mode on a 7-point Likert scale (Table 1, left). Participants rated all modes positively, but rated direct touch most highly. Given that participants were novices, it is possible that the more advanced features would be rated more highly after participants gained more experience. One participant, who was extremely technically savvy, praised the idea of the virtual edge menu as "kind of brilliant."

Document types: Participants also rated AL's usefulness for various document types (Table 1, right). Participants enjoyed the diagram and map tasks, but were less positive about using AL to navigate a table. During the study, participants sometimes had difficulty following the rows and columns of the table with their fingers. Several participants suggested adding a "table mode" that would provide feedback about the table structure and support traversing the table with gestures.

Interaction mode	Mdn. Rating	Document type	Rating
Direct touch	7	Diagram	5
Edge menus	6	Map	6
Voice commands	5	Table	4
Color recognition	7		

Table 1. Likert-scale responses (1=worst, 7=best): 1) preferred interaction mode; 2) best documents to use with Access Lens.

Usability and reliability challenges: All participants in the lab study were able to use AL to complete the tasks. However, participants did encounter some usability problems. Some participants had difficulty keeping track of the camera's view. Often, participants inadvertently moved the document as they were reading, causing AL to report text at incorrect locations. This problem could be addressed by securing the document to the surface, or by re-scanning the document. Participants also confused the gesture tracker by placing both hands on the document, or by holding their hands at an angle. AL's gesture tracker expected users to clearly extend the pointer finger, as in Figure 2. As a result, AL sometimes tracked the side of the participant's hand, rather than the finger. During the study, we reminded participants to extend their fingertip, although a more robust finger-tracking algorithm could also solve this issue.

A second question raised by this study is the degree to which AL is ready for real-world deployment. We found that AL's vision system performed reliably under varying indoor lighting conditions, but that camera settings sometimes required manual adjustment to ensure proper gesture tracking, which could present challenges when deployed to blind users in the wild. OCR generally took between 30 seconds and 1 minute. OCR accuracy varied greatly with lighting, camera settings, and camera position, but was typically well above 50%, even correctly recognizing proper names and numbers. However, these results are likely bound to our chosen lab setting, and results from the wild may differ significantly. Our planned field study, described below, will provide more information about the robustness of AL's vision system in the wild.

FUTURE WORK

The present work demonstrates the usability of AL in a lab setting. However, we intend to improve AL by adding new features and by increasing its robustness to real-world environmental conditions. One persistent challenge is the inevitable presence of OCR errors. To address this problem, we developed a crowdsourced OCR module based on QuikTurkIt [1]. Crowd OCR is currently slower than automated OCR, taking 2-3 minutes per page,[3] but is more accurate, especially when image quality is low. As there are performance, privacy, and cost tradeoffs between recognizers, we are developing an interface that will allow the user to select their preferred recognizer, and fall back to a secondary recognizer if the primary recognizer fails.

We also intend to conduct an extended field evaluation of AL. While AL performed well in the lab, field study participants will likely encounter environmental conditions that will negatively affect performance. Improving performance in these contexts may require better camera calibration algorithms, especially since blind participants may not be able to manually calibrate the camera themselves. This field study will help us to identify which types of objects users are most interested in scanning, which may allow us to optimize our recognition algorithms.

While we believe that AL provides valuable accessibility support in its current form, there is much potential in alternative form factors for AL, especially mobile form factors. We have constructed a wearable hardware prototype in the form of a pendant camera combined with an ultra-mobile PC. However, creating a reliable mobile version of AL will require further improvements to camera calibration and OCR. Furthermore, mobile AL will require a way to detect movement of the camera or scanned object, and to recalibrate the locations of the scanned text.

Finally, we are interested in extending AL to recognize additional content beyond text and color. Future versions could identify images or glyphs printed on maps, signs, or other artifacts. We have also experimented with using AL to read computer displays and non-accessible touch screens, and will further explore this use case in the future.

[3] Recognition time can likely be reduced via improved crowd algorithms.

CONCLUSION

While the rapid adoption of touch screen-based devices first seemed to be a threat to accessibility for blind users, the development of accessible gestures has helped to ensure that such devices remain accessible. Access Lens leverages advancements in accessible gestures to create a new form of assistive technology. While much of the previous research on accessible gestures has focused on providing access to specific technologies, our study shows that accessible gestures can also provide access to the physical world.

REFERENCES

1. Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Tararowicz, A. White, B., White, S., and Yeh, T. VizWiz: nearly real-time answers to visual questions. *Proc. UIST '10,* (2010), 333–342.
2. Bradski, G. The OpenCV library. *Doctor Dobbs Journal 25*, 11 (2000), 120–126.
3. Kane, S.K., Avrahami, D., Wobbrock, J.O., Harrison, B., Rea, A.D., Philipose, M., and LaMarca, A. Bonfire: a nomadic system for hybrid laptop-tabletop interaction. *Proc. UIST '09,* (2009), 129–138.
4. Kane, S.K., Bigham, J.P., and Wobbrock, J.O. Slide Rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. *Proc. ASSETS '08,* (2008), 73–80.
5. Kane, S.K., Morris, M.R., Perkins, A.Z., Wigdor, D., Ladner, R.E., and Wobbrock, J.O. Access Overlays: improving non-visual access to large touch screens for blind users. *Proc. UIST '11, (2011),* 273–280.
6. Levenshtein, V.I. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission 1*, 1 (1965), 8–17.
7. Mahmoud, T.M. A new fast skin color detection technique. *WEAST 43*, (2008), 501–505.
8. Nanayakkara, S., Shilkrot, R., and Maes, P. EyeRing: a finger-worn assistant. *Proc. CHI EA '12, (2012),* 1961–1966.
9. Stauffer, C. and Grimson, W.E.L. Adaptive background mixture models for real-time tracking. *Pro*c. *CVPR '00*, 252–258.
10. Sukthankar, R., Stockton, R.G., and Mullin, M.D. Smarter presentations: exploiting homography in camera-projector systems. *Proc. ICCV '01*, 247–253.
11. Wellner, P. Interacting with paper on the DigitalDesk. *Communications of the ACM 36*, 7 (1993), 87–96.
12. White, S., Ji, H., and Bigham, J.P. EasySnap: real-time audio feedback for blind photography. *Proc. UIST '10*, (2010), 409–410.
13. Wong, E.K. and Chen, M. A new robust algorithm for video text extraction. *Pattern Recognition 36*, 6 (2003), 1397–1406.