Leveraging Augmented Reality to Create Apps for People with Visual Disabilities: A Case Study in Indoor Navigation

Leave Authors Anonymous
for Submission
City, Country
e-mail address

Leave Authors Anonymous
for Submission
City, Country
e-mail address

Leave Authors Anonymous
for Submission
City, Country
e-mail address

ABSTRACT

The introduction of augmented reality technology to iOS and Android enables, for the first time, mainstream smartphones to estimate their own motion in 3D space with high accuracy. For assistive technology researchers, this development presents a potential opportunity. In this spirit, we present our work leveraging these technologies to create a smartphone app to empower people who are visually impaired to more easily navigate through indoor environments. Our app, *Clew*, allows users to record routes and then load them, at any time, providing automatic guidance (using haptic, speech, and sound feedback) along the route. We present our user-centered design process, Clew's system architecture and technical details, as well as both small and large-scale evaluations of the app. We discuss opportunities, pitfalls, and design guidelines for utilizing augmented reality for orientation and mobility apps. Our work expands the capabilities of technology for orientation and mobility that can be distributed at massive scale.

ACM Classification Keywords

K.4.2. Assistive technology for persons with disabilities: Orientation and mobility tools for persons with visual disabilities

Author Keywords

orientation and mobility; augmented reality; assistive tech

INTRODUCTION

For people who are blind or visually impaired (B/VI), improvements in orientation and mobility (O&M) have been shown to increase economic opportunity as well as psychological well-being. For instance, while only 30% of working-age Americans who are B/VI are employed [2, 26], individuals with better O&M skills fare better at finding employment [10, 9, 27, 35]. Due to the importance of O&M, there is a long history of assistive technologies designed to bolster these skills [3, 6]. Despite considerable effort, historically, few of these technologies have achieved much impact beyond the lab [39]. In part, this lack of impact has been driven by the fact that the technologies were expensive, unreliable, cumbersome, or hard to distribute at mass scale.

One notable exception to this disappointing track record are GPS navigation apps for smartphones. Since these apps are built on top of mainstream technologies that happen to be universally accessible — as opposed to special purpose assistive technologies — they are robust, powerful, and extremely useful to people who are B/VI. Further, the fact that a majority of people who are B/VI own smartphones [33] makes them distributable at either no additional cost (e.g., Google Maps) or at modest cost (e.g., BlindSquare, which is designed for people who are B/VI and costs about $30).

Despite the undeniable impact of smartphone-based GPS technology, importantly, mass market GPSes are only accurate to about $5m$ under open sky (and are even worse in challenging environments such as cities) and do not work indoors. These limitations make them inapplicable to both high precision navigation as well as navigation in indoor environments. In response to the need for technology to assist with these sorts of O&M tasks, researchers are working to build systems that overcome these challenges (see *Related Work*).

Recently, smartphone manufacturers have introduced augmented reality (AR) modules that support high-accuracy 3D-tracking. While the primary purpose of these modules is to enable AR experiences — whereby virtual and physical content are mingled, e.g., by overlaying virtual characters on a smartphone's camera feed — these modules have the potential to be repurposed to create assistive technology for O&M that is both high-precision and usable indoors. Further, since AR is now a standard feature of modern smartphones, O&M apps based on this technology can be widely deployed at no cost.

With the significant potential of smartphone-based AR technology comes critical research questions. Are the motion estimates provided robust enough to use for O&M? If so, which O&M tasks might be facilitated? What usability challenges does AR technology bring, and how can we, as designers, best support users in harnessing such technology? Here, we take preliminary steps towards answering these questions by presenting our work utilizing user-centered design to leverage the AR modules in modern smartphones to create, and to release to a large global audience, an application to assist with indoor navigation. Our app, *Clew*, enables users to record routes using their smartphone so that they can navigate these routes at a later point in time. *Clew* is designed to alleviate various

pain points experienced by folks who are B/VI (e.g., finding one's way back after being guided to a location or following a new route in an unfamiliar environment).

In the remainder of the paper we present related work on O&M assistive technology, discuss the algorithms that underlie AR technology on smartphones, present the design of *Clew*, provide small- and large-scale evaluations of *Clew*, and discuss future challenges and promising directions for smartphone-based AR technology for people who are B/VI.

RELATED WORK

A number of researchers have worked to overcome the limitations (e.g., less-than-ideal accuracy, lack of availability indoors) of GPS technology for assisting people who are B/VI with O&M. Researchers have pursued roughly two approaches. The first is to utilize crowdsourcing, whereby people who are B/VI connect over the internet with a sighted person for real-time assistance. Examples of this approach include the VizWiz project [4], BeMyEyes [23], and Aira [8] (both BeMyEyes and Aira support a video chat interface). A second approach combines rudimentary motion estimates derived from inertial sensors (gyroscopes and accelerometers) with detection of fixed environmental infrastructure (e.g., Bluetooth beacons, Wifi access points, infrared emitters). For instance, [19, 18, 20] developed a system for navigation using smartphone-detectable RFID tags. Dias and her collaborators utilized WiFi fingerprinting and dead-reckoning for indoor navigation [12], and similar systems based on Bluetooth beacons have also been developed [25, 1]. Others have explored the use of robots to act as guides for people who are blind [34].

There is some work on recording and retracing routes in unmodified indoor environments. In [14, 15], the authors create a system that counts steps (for estimating linear motion) and detects 90-degree turns to first record and then to retrace a route. In [17], the authors present a localization system based on combining AR-based tracking and sign recognition. This work did not test a full navigation system in that navigational guidance was provided by an experimenter that followed along behind the user. The authors found satisfactory localization performance of their system for two of the test users and poor performance for the third as measured along three indoor routes. In comparison to these two lines of work, the novelty of our work lies in support for route saving and reloading while making minimal assumptions about the environment (e.g., the availability of signs for tracking or that all turns are 90 degrees), a large-scale evaluation of our app, and the generation of specific design guidelines for researchers who would like to incorporate AR technology into O&M applications.

AUGMENTED REALITY

Both Apple and Google have released support for smartphone-based AR experiences, whereby virtual and real content are combined. For instance, an app might show a virtual cat projected into a real world scene. As the user moves, the phone senses the user's motion and renders the cat at an appropriate distance and angle, providing the illusion that the cat exists in the physical world.

These AR systems utilize 3D motion-tracking algorithms that combine data from inertial sensors (gyroscopes and accelerometers) with visual information (obtained from the phone's camera) to generate motion estimates that are far more accurate than what could be obtained using only inertial sensing. The high accuracy of these systems is enabled by two key trends: the development of sophisticated algorithms for visual-inertial odometry (VIO) [29, 28, 5, 16] (which are the algorithms that enable the combination of optical and inertial data for motion estimation) and the development of special-purpose hardware to allow computationally intensive algorithms to run with minimal heat generation and power consumption.

Algorithms for Visual Inertial Odometry

While a full explanation of VIO [21] is beyond the scope of this paper, it helps to have a conceptual understanding of VIO. VIO algorithms are designed for either the monocular (single camera) or stereo setting. Since the monocular setting is the one currently applicable to mass-market smartphones, here we use the term VIO to refer to monocular VIO specifically.

VIO algorithms utilize sensor fusion to blend optical and inertial motion estimates. Optical motion estimates are made by tracking salient visual features — e.g., corners or other textured portions of an image — through multiple video frames. Utilizing the mathematics of perspective geometry, one can estimate the rotation and translation of the camera [22]. Importantly, the accuracy of these estimates is dependent on tracking a large number of visual features that should, ideally, correspond to points at a range of depths from the camera.

Further complicating matters, the translation estimated using optical tracking is only determined up to an arbitrary scale factor. This indeterminacy arises due to the fact that the depths of the tracked visual features are unknown [22]. For example, given an estimate of the translation of a camera, it is possible that the camera moved twice as far and the depths of the tracked points were twice as great. The shortcomings of optical tracking, scale-indeterminacy and inaccurate performance in feature-poor environments, can be overcome by the fusion of inertial measurements from gyroscopes and accelerometers. Gyroscopes, which provide accurate estimates of angular velocity, can refine estimates of rotation while accelerometer data can be integrated over time to obtain estimates of linear velocity, overcoming the scale-indeterminacy problem.

VIO in Mass-Market Smartphones

Both Apple and Google have released AR modules based on VIO. While, the details of their algorithms are not publicly available, there are distinctions between these frameworks that researchers should keep in mind.

Google Tango

Release in late 2014 by Google's ATAP (Advanced Technology and Projects) division, Tango utilizes a wide-angle, fisheye lens and a global image shutter to enable accurate visual-feature tracking. The platform also includes a Prime-Sense depth-sensing camera. Two commercial smartphones have been released based on the Tango platform: the Lenovo Phab2 Pro and the Asus Zenfone AR. While the tracking capabilities of Tango devices are superior to both ARKit and

ARCore (discussed next), the reliance on special-purpose cameras severely limited the adoption of the technology. As a result, Google suspended the project in early 2018 [11].

ARKit and ARCore
Apple's ARKit [24], released in 2017, and Google's ARCore [30], released in 2018, do not require special-purpose cameras. Since these platforms utilize conventional cameras, which have comparatively limited fields-of-view when compared to a fisheye camera, the richness of visual features available for tracking is not as great as with Tango. Consequently, motion estimates are less accurate. Further, since neither of these platforms have depth sensing cameras, the availability of 3D information is limited to objects with special structure (e.g., horizontal and vertical planar surfaces). Despite their drawbacks, these frameworks can run on a wider array of phones than Tango, and given the large preference for iOS among people who are B/VI [33], ARKit is the primary platform of interest for researchers seeking to develop O&M apps for people who are B/VI.

DESIGN PROCESS OVERVIEW
We employed a user-centered design process that involves people who are B/VI in all phases (see Figure 1). In the initial phase, we focused on in-person interactions with local members of the B/VI community and members of the research team who are B/VI to identify areas of opportunity for improving access to physical spaces (see *User Interviews and Areas of Opportunity*). Based on these interactions, we developed concepts for O&M apps. These app concepts fed into an intensive co-design process where we created system designs, selected and developed algorithms, and ultimately produced prototype apps, which our co-designers engaged with (see *User Interviews and Areas of Opportunity*). We iterated this design cycle based on feedback from our co-designers.

Once we had an app that was polished as well as useful to the B/VI community, we released the app on the iOS App Store. The release of the app generated feedback in two forms. First, users from all over the world gave us their impressions of the app and how they would want to see it improved. Second, users who opted-in to sharing usage data, provided a rich dataset to understand, in a quantitative manner, how the app was being utilized (see *Large-Scale User Study*).

User Interviews and Areas of Opportunity
We employed semi-structured interviews and participatory design approaches [7, 37] with members of the local B/VI community. In the very earliest stages of the process, we conducted a series of co-designs with Joe (pseudonym), a college student who has no functional vision due to Retinitis Pigmentosa (a degenerative vision disorder that leads to the breakdown of cells in the retina). Joe is a non-visual traveler, mostly navigating with the help of his guide dog. We also talked at length with an O&M trainer for primary and middle school students who shared stories of mobility challenges that his students face. Furthermore, three members of our research team, two of whom are low-vision and one of whom is completely blind, contributed to the design and implementation of the app and leveraged their personal experiences and knowledge of the B/VI community to inform this work. Our team identified a number of pain points and areas of opportunity related to O&M and access to physical spaces.

Navigating in unfamiliar indoor environments is difficult. In these situations, folks will often need to either ask a sighted person to assist them or bring along a sighted friend or family member. If one will be navigating this environment over a long period of time, for instance when starting a new job, one will often work with a mobility instructor in order to learn how to effectively get around in this new environment. A common sentiment among our co-designs was that relying on others for assistance in these scenarios is a major impediment to independence.

Navigating newly traveled routes towards previously visited locations is difficult. A specific subset of the difficulties encountered in navigating new indoor environments is navigating back to a starting point after traveling to a new location. As an example, a member of our design team expressed that finding his seat (e.g., in a classroom) after going to the restroom was challenging. A second example is that one of our visually impaired team members found it difficult to find his seat on a dimly lit airplane when returning from the restroom.

The identification of these pain points motivated the creation of *Clew* to support indoor navigation in unfamiliar environments.

CLEW APP OVERVIEW
Clew is an iOS app built on the ARKit platform that provides continuous guidance to users who are B/VI as they travel along pre-recorded routes. The app serves two main use cases. The first use case is to allow the user to record a path through an indoor environment and then navigate the path back to the starting location. This use case can arise, e.g., when a user is led somewhere by a sighted guide and they want to return to their previous location — without being guided back. As a second example, sometimes it is easier to navigate from a location than back to it. For instance, it is easier for someone who is B/VI to leave a conference room than it is to find their way back to their particular seat. The second use case is to allow the user to record a path through an indoor environment, save this path, and then navigate the path either in the forward or reverse direction at a later point in time. This function is useful when a user is either learning to navigate a route and could use guidance when practicing or when the user wants to navigate a new route repeatedly for a brief period of time (e.g., during a hotel stay a user may want to be able to navigate to their room or to the hotel pool).

In order to serve each of these use cases, *Clew* provides the core functionality of route recording along with high-accuracy, easy-to-follow, navigational guidance to enable people who are B/VI to travel independently indoors, without the need for modifications to the environment (e.g., the introduction of beacons or special signage).

Path Recording
In path recording mode, the app lays down a trail of virtual breadcrumbs (representing timestamped, 3D positions and
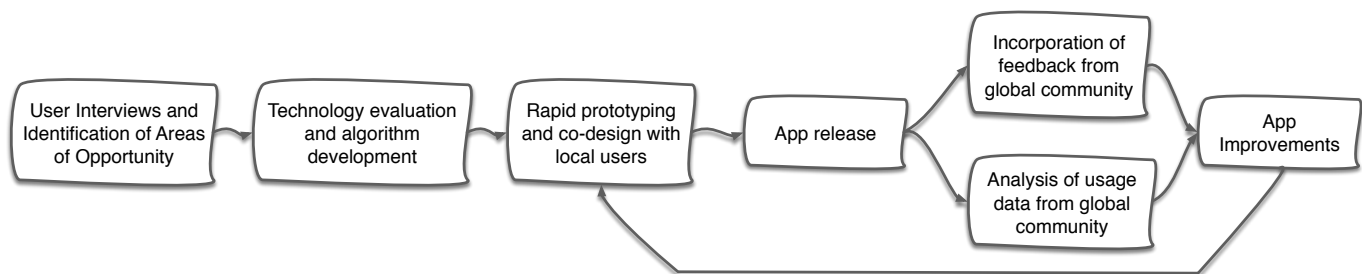
Figure 1. Our design process used to investigate the usage of AR technology for creating assistive technologies for people who are B/VI. We are using this process to design our app *Clew*.
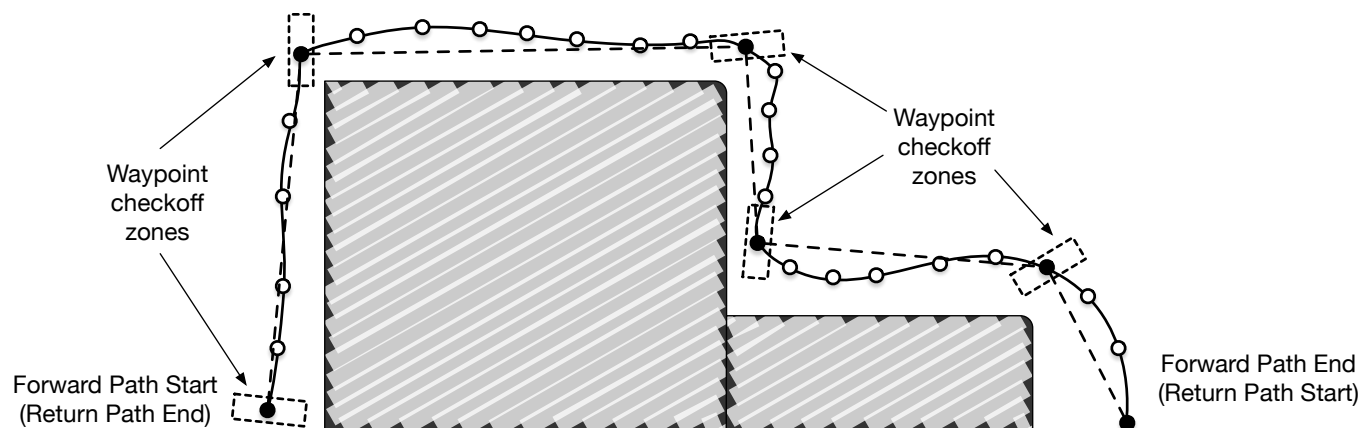

Figure 2. A topdown view of a path generated by the *Clew*. The figure shows the raw path (solid line) and breadcrumbs (white circles with black outlines). The Ramer-Douglas-Peucker reduces the breadcrumbs to a more manageable number of waypoints (solid black circles). The resultant path consists of straight lines connecting these waypoints in reverse order. Waypoint checkoff zones are shown as dashed boxes).

orientations) while the user traverses a route. These estimates of position and orientation are generated by Apple's ARKit, which provides these at a rate of 60 Hz. Recall that these position and orientation estimates are based on fusing visual and inertial measurements.

Once the user starts recording, they travel to a new location (either via their traditional O&M process or via assistance from a sighted guide) and then stop the recording. Figure 2 shows a sample path along with virtual breadcrumbs.

Route Pausing and Saving through Landmark Creation
ARKit tracks motion in a coordinate system whose origin coincides with the device's position when the app starts. Thus the positions of any breadcrumbs dropped during the path recording phase are meaningful only *within* the context of the current ARKit tracking session. Once a path is recorded, if the user doesn't switch to a different app, lock the phone, or occlude the camera, then they can navigate back to their starting location using the procedures described in *Path Navigation Mode*. However, if the user wishes to wait a significant amount of time before navigating back, would like to use another app, or if the user would like to save the route for use at a later time, these limitations can be prohibitive. In order to support such cases, Clew provides support for route pausing and saving through coordinate system registration.

Registration through Visual Alignment
With the release of iOS 12.0, Apple's ARKit supports a re-localization feature whereby a visual representation of landmarks and their associated 3D positions forms a sparse map of an ARKit session. Given a new ARKit session (e.g., the user has restarted the app), ARKit can load the previous 3D map and attempt to match the phone's current image features to the map. If a match is found, the current position and orientation of the phone are updated to be relative to the coordinate system in the 3D map (thereby registering the two coordinate systems). In order to support route saving and pausing, *Clew* saves the current 3D map of the tracking session and attempts to use visual alignment to relocalize the user when the user wants to reload the route.

The main limitation of visual alignment is that it doesn't always succeed. This can occur for a multitude of reasons. First, if the environment has changed significantly, for instance when lighting conditions have changed drastically, the visual environment may not look similar enough to the saved 3D map. Alternatively, even if the environment is static, the user's viewing angle might be significantly different. This is particularly likely to happen when navigating routes in the reverse direction as visual features in the environment often appear very differently when viewed from the opposite direction.

Registration through Physical Alignment
In order to allow users to reload or pause routes when visual alignment is impossible, Clew supports physical alignment
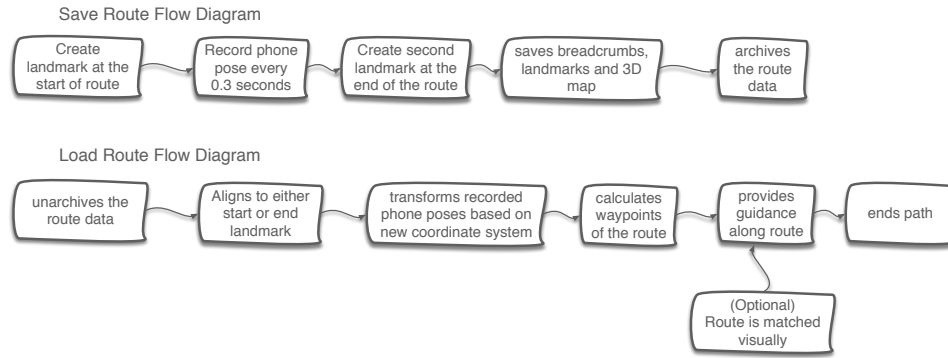
Save Route Flow Diagram

Create landmark at the start of route → Record phone pose every 0.3 seconds → Create second landmark at the end of the route → saves breadcrumbs, landmarks and 3D map → archives the route data

Load Route Flow Diagram

unarchives the route data → Aligns to either start or end landmark → transforms recorded phone poses based on new coordinate system → calculates waypoints of the route → provides guidance along route → ends path

(Optional) Route is matched visually

Figure 3. A flow diagram of the process for both saving and loading routes in *Clew*.

Continue straight for 11 feet

Continue straight and proceed downstairs
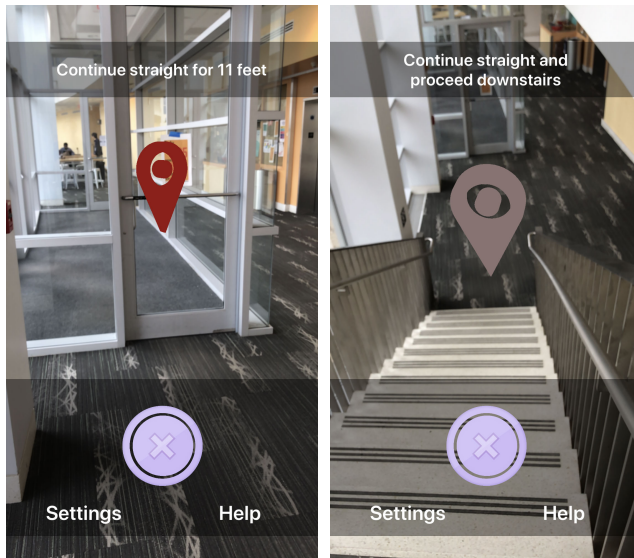
Settings   Help

Settings   Help

Figure 4. Two screenshots from the app "Clew." The text for the left image says "Continue straight for 11 feet" and the text on the right says "Continue straight and proceed downstairs."

through landmarks. A user can create a landmark by placing their phone in a position and orientation (collectively called a "pose") that is easy to return to, unassisted, when the user wants to resume or reload the route. The most important aspect of the pose that the user must successfully recreate is its yaw. The reason for this is that the phone's accelerometer can determine the other two degrees of freedom of rotation, pitch and roll, as these are not perpendicular to gravity and deviations in the phone's position contribute a fixed amount of error, whereas an error in the phone's yaw will be magnified over long routes.

The procedure for landmark creation is for the user to place their phone's top (short-edge) flush against a flat vertical surface (such as a door or wall). In this configuration the user's camera will be facing down, which will allow it to track visual features on the floor, and the screen will be facing up, which allows the user to see what's on the screen (if they have usable vision) or interact with the phone using VoiceOver. In order to make the alignment process more robust, we artificially "level" the phone by undoing the effects of roll and pitch to create

a virtual landmark pose in which the phone is perfectly flat. We found that this leveling process was crucial to maximizing accuracy. Clew allows the user to enter text or record a voice message to help them remember the details of how they positioned their phone when creating the landmark (e.g., "Office front door, right above the handle"). While in theory we could support registration with visual alignment only, we made the design decision to require the user to physically register a landmark in order to reload or pause a route. An flow diagram of the saving and loading procedures is shown in Figure 3.

Path Navigation Mode
When the user is ready to navigate a route, the trail of breadcrumbs is processed by the Douglas-Ramer-Peucker (DPR) algorithm [13] for path simplification. This algorithm winnows down the path by removing sequences of breadcrumbs that are well represented by a straight line. Figure 2 shows the breadcrumbs selected by the DPR algorithm, called *waypoints*, and the resultant piecewise straight navigation route obtained by connecting the waypoints. In *navigation mode*, the app synthesizes directions to the next waypoint using one of three mechanisms: (1) speech (e.g., "continue straight for 10 feet"), (2) haptic or (3) audible feedback when the phone is pointing towards the next waypoint. When using (2) or (3), the low latency of the update of the phone's position allows the user to sweep their phone back and forth until they sense a haptic or auditory cue, providing an accurate sense of the direction to the next waypoint.

As the user navigates, the app continuously checks to see if the user has reached the next waypoint. We define the condition of "reached the next waypoint" as the user entering a *waypoint checkoff zone* (see Figure 2). Instead of using spherical checkoff zones, we use rectangular prisms whose sides perpendicular to the direction of travel are longer than for those parallel to the direction of travel. This choice of shape enables the user to deviate laterally from the intended path without missing a waypoint. The app also announces flights of stairs by detecting if the vertical angle of the segment connecting two waypoints exceeds a threshold.

USABILITY TESTING AND TECHNOLOGY EVALUATION
Previously, we discussed our initial design work that we undertook to inform the concept of Clew. Here, we discuss the

results of co-designs as well as data analyses that informed the creation of specific aspects of Clew.

Longitudinal Design with Local Co-Design Partners

We engaged with four local partners to co-design Clew. We worked with each co-designer for five, two-hour sessions spaced a week apart. Our co-designers brought useful insights along a range of dimensions. Here we highlight several that are of particular relevance to using AR technology for people who are B/VI.

Insight 1: Maintaining Optimal Phone Orientation

VIO algorithms work best when they detect visual features at a range of depths. The reason for this is that many of these algorithms perform geometrical calculations which are best numerically conditioned when visual features are tracked at a range of depths from the camera. This suggests that users should hold their phone upright with the camera facing approximately parallel to the ground. If the user's phone is pointed down at the floor or to the side at a wall, the phone will track features that primarily lie on a single plane, which will result in less reliable estimates of motion. We found that Joe had difficulty maintaining the phone in this configuration (perhaps due to the lack of visual feedback about the phone's orientation). As a result, tracking performance suffered. In our large-scale evaluation, we were able to demonstrate this quantitatively. Developers of AR-powered assistive apps may consider adding feedback mechanisms to help the user to maintain their phone in a vertical orientation.

Insight 2: Alignment Between Body and Phone

Through interactions with Jim (pseudonym), a local co-design partner, we discovered that some users have a difficult time understanding the orientation of their phone relative to the orientation of their body. Jim is congenitally blind, whereas the other three local co-designers, none of whom had this difficulty, lost their vision in their teens. That the age at which sight was lost would have this effect was not wholly unexpected (see, e.g., [31, 39, 36, 38, 40] for discussions of how spatial processing differs based on when the onset of vision loss occurred). We found that Jim could easily rotate his body in an effort to elicit haptic guidance from the app, which would indicate that he was facing the correct direction. He was not able to rotate the phone in his hand (with his body stationary), sense that the phone was pointing in the correct direction, and then align his body with the phone's direction.

In order to provide feedback relative to his body direction rather than the phone's direction, we developed a feature to constantly update an estimate of the phone's offset relative to the user's body. The calculation of this offset is performed when the user is moving forward (moving laterally will throw this calculation off). We found that this modification enabled Jim to use the app effectively. Developers of AR-powered assistive apps should be cognizant of the differing abilities of users to sense spatial relationships between various parts of their body and their phone.

Quantitative Evaluation of ARKit's Accuracy

In order to better understand the accuracy of ARKit for navigation, we performed a benchmarking experiment to test ARKit's

Route Length	Contains Staircase	Mean Error
10m	yes	0.27m
13m	no	0.51m
26m	yes	0.74m
38m	no	0.56m
63m	yes	0.50m

Table 1. Accuracy of relative position estimates for the start and end of the route. This number is indicative of the navigation accuracy that could be expected when using Clew over a similar route.

performance along five indoor routes of varying length and complexity. For each route, the phone was placed flush against a wall in a particular starting location. Once the phone was positioned properly, the *landmark procedure* (described earlier) was performed. Next, the experimenter walked the route (using their vision) to the route's endpoint where the landmark procedure was performed a second time. The relative position of the ending landmark to the beginning landmark provided an estimate of the spatial relationship between these two points.

To get a sense of how repeatable the estimate of the spatial relationship between the start and end of the route was, we repeated this process at least seven times per route. The error associated with both the physical alignment procedure (landmark creation) as well as the drift incurred by ARKit while navigating the route was quantified by calculating the mean distance between the average of the relative positions between the landmarks and the relative position for each trial. This metric provides a sense of how accurately ARKit could give feedback to the user regarding how close they are to the route's endpoint. While it does not say anything about the accuracy of ARKit in the middle of the route, the endpoint is likely to be where the error is greatest (as it is the farthest from the start and thus motion tracking error has had longer to accumulate).

The characteristics of the routes tested as well as the accuracy of the relative start-to-end position estimates are shown in Table 1. Even over routes as long as 63m, error is under 1 meter. This holds true when the route involves ascending or descending stairs. These results support the idea that ARKit's tracking performance is accurate enough for use in guiding users along recorded indoor routes of moderate length.

Feedback from Global User Community

In order to achieve impact on a large-scale and to understand the performance of our app at a fine-grained, quantitative level, we released *Clew* on the iOS app store in 156 countries. In this section, we summarize the most important lessons we learned from the global community as to how to utilize AR technology to create assistive technology for people who are B/VI.

Adoption and Usage of Clew

We initially promoted the app on a limited basis through word-of-mouth to local members of the B/VI community. Several members of the community discovered Clew and created posts on the popular assistive technology portal *Applevis*.

The publicity created by the *Applevis* posts generated a number of additional articles and blog entries. As of the submission of this paper, Clew has been downloaded by almost 4,000 users

from 50 different countries. Clew is used by an average of 80 people per day.

Designing for Low-vision Users
One somewhat surprising thing we heard from the community is that Clew was useful for folks who are low-vision (rather than just folks who are blind). In our design of the initial version of Clew, we had focused on making Clew accessible via VoiceOver. We didn't put much intentionality into the visual design of the app for folks with low vision. When the app began to gain popularity we received feedback that the app's text was too small, the visual design of the waypoint markers was suboptimal, and that our speech guidance did not work if VoiceOver was turned off. We addressed many of these concerns in subsequent updates.

Supporting Older iOS versions
In response to a beta version of the app that removed support for an older version of iOS, one user expressed a desire to maintain this support. We were puzzled by this request as we assumed that all users would have upgraded to the most recent iOS (which was 12.2). The user explained that he didn't update his phone until the new major revision had been out for a significant amount of time. He went on to say that the minor bugs that often accompany major releases that sighted users can workaround can be showstoppers for folks that rely on the accessibility features of iOS. As a result, we reversed course and maintained support for older iOS versions.

Importance of Internationalization and Localization
The majority of the users of Clew are from outside of the U.S. The fact that Clew was released, and is still available only in English, is a major limiting factor in the adoption and successful usage of Clew. We have received many requests for Clew to be localized into other languages as well as a number of offers to perform localization work pro bono. The fact that language barriers created a lack of accessibility for an app designed to promote indoor accessibility was an irony we had not thought about ahead of time.

Large-Scale User Study
After the user successfully completes the navigation of a route or cancels navigation, they are prompted to rate the quality of the app's navigational guidance as either "thumbs up" or "thumbs down." While the exact meaning of these ratings is subjective (we don't provide any guidance to the user about what these mean), it does provide us with data to understand the factors that influence the quality of the user's experience. Importantly, users could opt out of contributing their route ratings and log data, so we only have access to data that represents a subset of the overall usage of the app.

In order to investigate the association between various factors and the user's rating of a navigation experience, we performed a logistic regression analysis with route rating as the dependent variable and the following independent variables.

- a binary feature that indicated whether or not the route was resumed/reloaded or newly recorded

- a binary feature that indicated whether the route was in the same direction of the recording or in the reverse direction
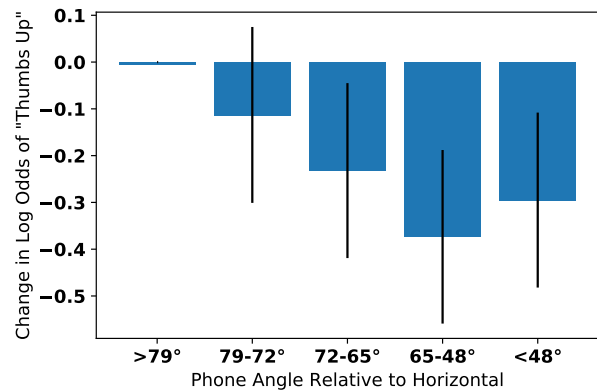
Figure 5. The relationship between log odds of the user rating a navigation experience as "thumbs up" and the median angle of the phone relative to horizontal (i.e., a plane perpendicular to gravity). Also shown are the 95% confidence intervals.

- a categorical feature that indicated which of five percentile bins the length of the route fell within (0-20th percentile, 20th-40th percentile, 40th-60th percentile, 60th-80th percentile, 80th-100th percentile).

- a categorical feature that indicated the number of motion racking errors reported by ARKit during the route navigation (0, 1, 2, 3, or more than 4).

- a categorical feature that indicated the median angle of the user's device relative to the ground plane during route navigation. The feature indicated which of five percentile bins the median device angle fell within (0-20th percentile, 20th-40th percentile, 40th-60th percentile, 60th-80th percentile, 80th-100th percentile).

In total, our analysis covered 5,789 routes. A total of 490 of these routes were collected after we released support for route saving and reloading (in the first few versions of Clew, route pausing was supported but not route reloading). The baseline "thumbs up" rating in the dataset was 68%.

Importance of Device Pose During Navigation
As mentioned previously, we knew from both the underlying details of the algorithms and from our anecdotal observations that VIO performed sub-optimally when the phone was not held vertically. Further, we had seen that some of our co-designers had difficulty keeping the phone vertical while using the app. The regression analysis showed that, consistent with these observations, users were less likely to rate their navigation experiences as positive when the phone was held at a flatter angle (see Figure 5). This finding further underscores the utility of providing feedback to help the user maintain their phone's vertical orientation when using AR applications.

Importance of Minimizing Tracking Failures
ARKit reports two error conditions related to motion tracking. One error condition is "insufficient visual features," which occurs when the camera is not capturing images with sufficient texture to allow for accurate feature tracking. The second error condition is "excessive motion," which occurs when the
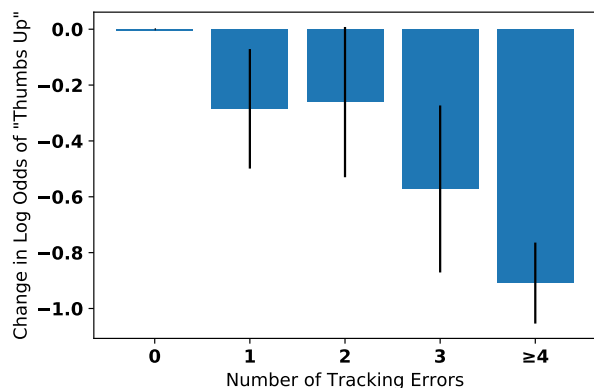
Figure 6. The relationship between log odds of the user rating a navigation experience as "thumbs up" and the number of tracking errors encountered during route navigation. Also shown are the 95% confidence intervals.

Figure 7. The relationship between log odds of the user rating a navigation experience as "thumbs up" and the length of the route. Also shown are the 95% confidence intervals.

phone is moving too quickly. Our analysis found that there was a steep drop-off in the likelihood of the user rating the navigation experience positively as the number of tracking errors increased (see Figure 6). This finding prompted us to change the behavior of the app to announce tracking errors to the user in hopes that this feedback would help the user to avoid them.

Route Characteristics and Average Rating
In addition to the quality of the tracking of the user's phone during navigation, we also found that various characteristics of the route itself were related to the likelihood of the user providing a positive rating to their navigation experience.

Routes that were resumed (versus newly recorded) had a lower log-likelihood ratio of being rated positively ($p < 0.002$). This finding is consistent with the inaccuracies that can be introduced by Clew's route alignment procedure. Routes that were resumed but navigated in the forward direction had a higher likelihood of being rated positively, but the result was not statistically significant ($p = 0.2$). If this finding holds, this is likely related to the higher success rate of visual alignment when navigating in the forward direction where visual features are more similar to those seen during route recording.

The relationship between route length and likelihood of a positive rating had an unexpected form (see Figure 7). The figure shows that the navigation experiences that are most likely to be rated positively are for routes that are between 9 and 15 meters in length. We had expected that the likelihood of a positive rating would monotonically decrease with route length. We think this is due to the fact that users are forced to rate every route and have no way to cancel navigation without issuing a rating. The lower ratings for shorter routes could be due to users wanting to cancel route recording and being forced to issue a rating of a route they didn't intend to record.
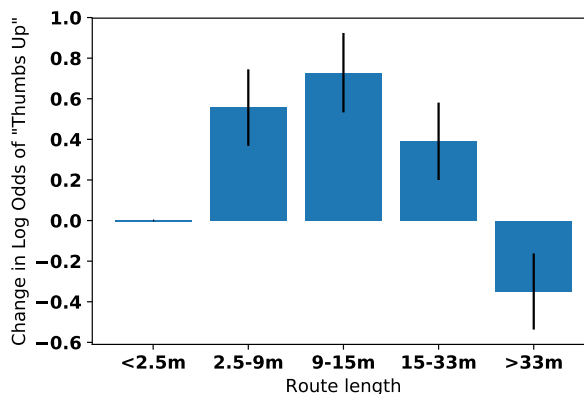
SUMMARY OF CONSIDERATIONS FOR RESEARCHERS WHO WANT TO USE SMARTPHONE-BASED AR FOR O&M
Based on our co-design process, technology evaluation, and user study, we summarize our advice for assistive technology researchers who want to use AR for O&M as follows.

- *Without any additional environmental instrumentation, ARKit is robust enough for many indoor navigation scenarios.* Our study demonstrates that for navigation routes of $\sim 61m$ the motion estimates of ARKit are accurate.

- *Maintaining the phone in a vertical orientation is important for optimal performance.* It is an open question as to how to incorporate this finding into the design of the user experience of an app (e.g., whether it is a good idea to provide explicit feedback when the phone is too flat).

- *Design for multiple levels of vision.* While it is tempting to think of apps in this space as special-purpose assistive technology, based on our experience we feel that taking a universal design approach will make the end product useful to folks who you didn't imagine were part of your user group (e.g., users who are low-vision).

- *Consider differing abilities to process spatial information.* In addition to the spectrum of visual abilities of users, users will have a spectrum of ability to understand spatial relationships between various parts of their body and their phone. Providing mechanisms to alleviate the need for the user to have a highly precise sense of space will make an app more widely usable.

- *Consider investing in internationalization and localization upfront.* Since blindness is a low-incidence impairment, to get a broad sample of users may require working with people at a global scale. Releasing an English-only app sends a message of exclusion to would-be co-designers.

- *Design your interface with distributed co-design in mind.* The feedback mechanism of "thumbs up" versus "thumbs down," provides a relatively narrow lens to understand the user experience using the app. Consider building in a richer

set of feedback mechanisms to gather information from users about how your AR-powered app is functioning for them in their particular environment. Thinking of those that use your app as co-designers, rather than testers, may help in framing how you think of these interfaces.

FUTURE WORK AND AREAS OF OPPORTUNITY
While we have provided initial steps towards understanding and realizing the potential of AR technology for O&M, we have identified a number of broad future directions that are outside the scope of our current work.

3D Object Localization
Joe mentioned that he had no difficulty in finding objects in spaces he has control to organize himself, he admitted that locating objects of interest in unfamiliar places or when others disturb his organization can be very challenging. ARKit provides support for 3D localization of points that are located on horizontal or vertical planes. Further, if a point can be found in multiple views (e.g., using an automatic object localization neural network), triangulation can be used to recover the 3D position. We are currently working on an app that helps with finding lost objects using these ideas (i.e., object recognition, object localization using neural networks, and 3D position estimation using ARKit).

AR in Outdoor Environments
While the focus of our work is navigation within indoor environments, in theory Clew can be used outdoors (we have done some testing of this). Outdoor navigation presents some challenges for AR. Specifically, visual features are often either far in the distance, which doesn't help with translation estimation, or located on a plane, which can lead to poor numerical conditioning. Integrating AR and GPS might help address this problem. Importantly for researchers to keep in mind, Google is beta testing an augmented reality-based version of Google Maps with a subset of its users [32].

Enabling Hands-free Operation
Future research should consider whether a method of hands-free operation could be developed that allows the user to continue to use their phone as an interface. This could work by using an attachment (e.g., a neck lanyard) for holding the phone coupled with a speech-based user interface.

CONCLUSION
We have presented our smartphone app, Clew, which allows people who are B/VI to record and navigate along routes. Our app is among the first mass-distributable apps for navigation in unmodified indoor environments.

Additionally, we have outlined areas of opportunity for the development of AR apps for O&M and provided a discussion of guidelines for successfully using this technology. Through the development of new algorithms, co-design with users, and the improvement of AR technology itself, researchers will hopefully succeed in leveraging AR to create impactful smartphone technology for people who are B/VI.

REFERENCES
1. Dragan Ahmetovic, Cole Gleason, Chengxiong Ruan, Kris M Kitani, Hironobu Takagi, and Chieko Asakawa. 2016. NavCog: a navigational cognitive assistant for the blind.. In *MobileHCI*. 90–99.

2. American Federation for the Blind. 2017. Interpreting Bureau of Labor Statistics Employment Data. (January 2017). `http://www.afb.org/info/blindness-statistics/interpreting-bls-employment-data/24`

3. J Malvern Benjamin. 1973. The new C-5 laser cane for the blind. In *Proc. Carnahan Conf. on Electronic Prosthetics*. 77–82.

4. Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and others. 2010. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 333–342.

5. Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. 2015. Robust visual inertial odometry using a direct EKF-based approach. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 298–304.

6. Johann Borenstein and Iwan Ulrich. 1997. The guidecane-a computerized travel aid for the active guidance of blind pedestrians. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, Vol. 2. IEEE, 1283–1288.

7. Christian Bühler. 2001. Empowered participation of users with disabilities in R&D projects. *International Journal of Human-Computer Studies* 55, 4 (2001), 645–659.

8. Aira Tech Corp. 2018. aira: your life, your schedule, right now. `https://aira.io/`. (2018).

9. Adele Crudden and Lynn W McBroom. 1999. Barriers to employment: A survey of employed persons who are visually impaired. *Journal of Visual Impairment and Blindness* 93 (1999), 341–350.

10. Adele Crudden, Lynn W McBroom, Amy L Skinner, and J Elton Moore. 1998. Comprehensive Examination of Barriers to Employment among Persons Who Are Blind or Visually Impaired. *Mississippi State: Rehabilitation Research and Training Center on Blindness and Low Vision, University of Mississippi.* (1998).

11. Tech Crunch. 2017. Android Guys. `http://www.androidguys.com/2017/08/30/google-rebrands-tango-as-arcore/`. (August 2017).

12. M Bernardine Dias . 2014. *NavPal: Technology Solutions for Enhancing Urban Navigation*. Technical Report CMU-RI-TR-21. Robotics Institute, Pittsburgh, PA.

13. David H Douglas and Thomas K Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10, 2 (1973), 112–122.

14. German Flores and Roberto Manduchi. 2018. Easy Return: An App for Indoor Backtracking Assistance. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 17.

15. German Flores, Roberto Manduchi, and Enrique D Zenteno. 2014. Ariadne's thread: Robust turn detection for path back-tracing using the iPhone. *Proceedings of the IEEE Ubiquitous Positioning Indoor Navigation and Location Based Service* (2014).

16. Christian Forster, Matia Pizzoli, and Davide Scaramuzza. 2014. SVO: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 15–22.

17. Giovanni Fusco and James M Coughlan. 2018. Indoor Localization Using Computer Vision and Visual-Inertial Odometry. In *International Conference on Computers Helping People with Special Needs*. Springer, 86–93.

18. Aura Ganz, Siddhesh Rajan Gandhi, James Schafer, Tushar Singh, Elaine Puleo, Gary Mullett, and Carole Wilson. 2011. PERCEPT: Indoor navigation for the blind and visually impaired. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 856–859.

19. Aura Ganz, James M Schafer, Yang Tao, Larry Haile, Charlene Sanderson, Carole Wilson, and Meg Robertson. 2015. PERCEPT based interactive wayfinding for visually impaired users in subways. *Journal on Technology & Persons with Disabilities* 3, 22 (2015).

20. Aura Ganz, James M Schafer, Yang Tao, Carole Wilson, and Meg Robertson. 2014. PERCEPT-II: Smartphone based indoor navigation system for the blind. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE, 3662–3665.

21. Jianjun Gui, Dongbing Gu, Sen Wang, and Huosheng Hu. 2015. A review of visual inertial odometry from filtering and optimisation perspectives. *Advanced Robotics* 29, 20 (2015), 1289–1301.

22. R. I. Hartley and A. Zisserman. 2004. *Multiple View Geometry in Computer Vision* (second ed.). Cambridge University Press, ISBN: 0521540518.

23. Bill Holton. 2015. A Review of the Be My Eyes Remote Sighted Helper App for Apple iOS. *Access World Magazine* 16, 2 (2015).

24. Apple Inc. 2018. ARKit Apple Developer. `https://developer.apple.com/arkit/`. (2018).

25. Tatsuya Ishihara, Jayakorn Vongkulbhisal, Kris M Kitani, and Chieko Asakawa. 2017. Beacon-Guided Structure from Motion for Smartphone-Based Navigation. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 769–777.

26. Corinne Kirchner, Emilie Schmeidler, and Alexander Todorov. 1999. Looking at Employment through a Lifespan Telescope: Age, Health, and Employment Status of People with Serious Visual Impairment. *Mississippi State, MS: Rehabilitation Research and Training Center on Blindness and Low Vision.* (1999).

27. Robin Leonard, Tana D'Allura, and Amy Horowitz. 1999. Factors associated with employment among persons who have a vision impairment: A follow-up of vocational placement referrals. *Journal of Vocational Rehabilitation* 12, 1 (1999), 33–43.

28. Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. 2015. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research* 34, 3 (2015), 314–334.

29. Mingyang Li and Anastasios I Mourikis. 2013. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research* 32, 6 (2013), 690–711.

30. Google LLC. 2018. ARCore Overview. `https://developers.google.com/ar/discover/`. (2018).

31. Richard G Long and EW Hill. 1997. Establishing and maintaining orientation for mobility. *Foundations of orientation and mobility* 1 (1997).

32. Mashable. 2019. What it's like to walk with Google Maps in augmented reality. `https://mashable.com/article/google-maps-ar-augmented-reality-walking-navigation/`. (2019).

33. John Morris and James Mueller. 2014. Blind and deaf consumer preferences for android and iOS smartphones. In *Inclusive designing*. Springer, 69–79.

34. Amal Nanavati, Xiang Zhi Tan, and Aaron Steinfeld. 2018. Coupled Indoor Navigation for People Who Are Blind. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI '18)*. ACM, New York, NY, USA, 201–202. DOI: `http://dx.doi.org/10.1145/3173386.3176976`

35. Bonnie O'Day. 1999. Employment Barriers for People with Visual Impairments. *Journal of Visual Impairment & Blindness* 93, 10 (1999).

36. Victor R Schinazi, Tyler Thrash, and Daniel-Robert Chebat. 2016. Spatial navigation by congenitally blind individuals. *Wiley Interdisciplinary Reviews: Cognitive Science* 7, 1 (2016), 37–58.

37. Douglas Schuler and Aki Namioka. 1993. *Participatory design: Principles and practices*. CRC Press.

38. Catherine Thinus-Blanc and Florence Gaunet. 1997. Representation of space in blind persons: vision as a spatial sense? *Psychological bulletin* 121, 1 (1997), 20.

39. William R Wiener, Richard L Welsh, and Bruce B Blasch. 2010. *Foundations of orientation and mobility*. Vol. 1. American Foundation for the Blind.

40. Michele A Williams, Caroline Galbraith, Shaun K Kane, and Amy Hurst. 2014. just let the cane hit it: how the blind and sighted see navigation differently. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*. ACM, 217–224.