An automated AR-based annotation tool for indoor navigation for visually impaired people

PEI DU*, Portland State University, USA
NIRUPAMA BULUSU, Portland State University, USA

Low vision people face many daily encumbrances. Traditional visual enhancements do not suffice to navigate indoor environments, or recognize objects efficiently. In this paper, we explore how Augmented Reality (AR) can be leveraged to design mobile applications to improve visual experience and unburden low vision persons. Specifically, we propose a novel automated AR-based annotation tool for detecting and labeling salient objects for assisted indoor navigation applications like NearbyExplorer. NearbyExplorer, which issues audio descriptions of nearby objects to the users, relies on a database populated by large teams of volunteers and map-a-thons to manually annotate salient objects in the environment like desks, chairs, low overhead ceilings. This has limited widespread and rapid deployment. Our tool builds on advances in automated object detection, AR labeling and accurate indoor positioning to provide an automated way to upload object labels and user position to a database, requiring just one volunteer. Moreover, it enables low vision people to detect and notice surrounding objects quickly using smartphones in various indoor environments.

CCS Concepts: • **Human-centered computing → Accessibility**; *Ubiquitous and mobile computing*; Interaction design.

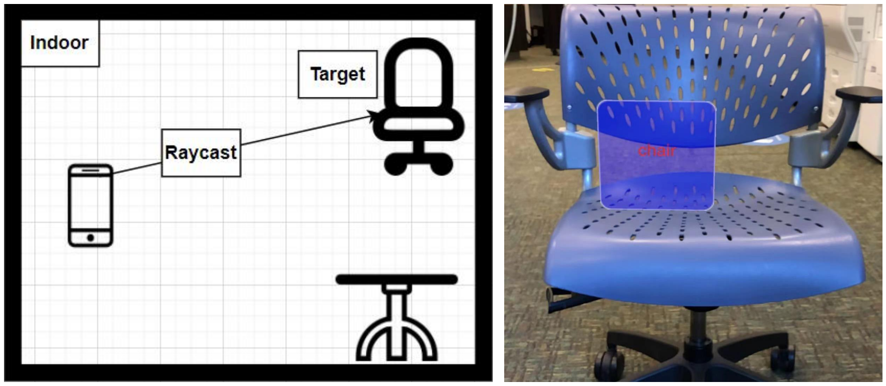
Additional Key Words and Phrases: augmented reality, indoor navigation, object detection, database

ACM Reference Format:
Pei Du and Nirupama Bulusu. 2021. An automated AR-based annotation tool for indoor navigation for visually impaired people. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '21), October 18–22, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3441852.3476561

1 INTRODUCTION

Visual impairment, a pervasive disability, includes limited vision, blurry vision, light sensitivity and blind spots[1]. In the United States, 19 million people are vision-impaired, encountering daily encumbrances in activities such as indoor wayfinding, relying traditionally upon audio and visual guidance [2–9]. However, these techniques do not always help them wayfind and identify objects reliably [10; 11]. For visually impaired people, the American Printing House for the Blind (APH) built a mobile application named Nearby Explorer [12]. For pilot indoor locations like the Millar Library at Portland State University, Nearby Explorer organizes periodic map-a-thons with university partners [10], recruiting and training large volunteer teams to manually enter environmental object data in appropriate annotation formats. This crowd sourced data, as in other crowd-sourcing efforts [13; 14], and complementary to personal object recognition methods [15], subsequently provides environmental cues during navigation to visually impaired people. Intensive effort makes it challenging to extend this greatly needed technology to other buildings.

We propose, design and implement a system to automatically collect data for Assisted Navigation Systems, replacing human effort with machine learning, augmented reality and indoor positioning technologies. This is non-trivial as it requires solving several practical challenges in minimizing duplicate object detections, automatically generating and



(a) (b)

Fig. 1. (a) Raycast, the iOS device emits a raycast that collides with target and returns the position. The user can affix a label on the position. (b) Example of an AR label.

recording object labels, and automatically associating object location with low effort. Our system uses YOLOv3, a highly accurate object detection algorithm and ARKit to identify objects with AR labels on Apple's iOS device. We also use IndoorAtlas, a SDK and cloud service which enable iOS and Android apps to acquire accurate end user position inside buildings near annotated objects. Relative to other indoor positioning systems, IndoorAtlas is more readily deployable given building floor plans which are usually available for North American buildings.

The contributions of our paper are two-fold. With our system, one volunteer who moves around and clicks buttons can accomplish data collection, greatly reducing both person-hours effort and cognitive load. Moreover, the AR annotation tool enables people with vision impairment to detect surrounding objects quickly and clearly using smartphones in various indoor environments. We next describe our work in object detection with AR labels (Section 2), a system built on this annotation tool (Section 3) and proof-of-concept results from our test environment, the Millar library (Section 4).

2 AR-BASED ANNOTATION

To realize AR-based annotation, we combine an object detection algorithm with the ARKit framework[16]. We train a custom YOLOv3 model [17] to automatically detect only objects related to wayfinding in our test environment, the Millar Library. Our mobile iOS app uses this model to automatically generates an AR label if object detection confidence exceeds a threshold. To create an AR label, we use SceneKit to establish an AR Scene, and the SCNNode class to represent a position and transform to a 3D coordinate space. App developers can attach geometry, lights and other displayable content to it. In the main screen, we designed an ARSCNView to display camera tracking status and a button to submit labels with timestamps to the database. The iOS device updates UI to provide feedback on the state of the AR experience. Only when tracking status is normal, a user can proceed to detect objects. If tracking status is not available or limited, the user should follow on screen instructions to adjust status. For example, users cannot move their device too fast, slowing down their device will reset the tracking status to normal. The users can hold the iOS device, rotate the device and move around to detect a surface up to a maximum distance of 0.5 m. When an object is automatically detected with confidence exceeding a threshold, it permits users to add annotation. To avoid repeated annotation for the same object, we develop a method building on HitTest method in ARKit. It searches the scene for objects corresponding to a point in rendered image. If the object has the same name as another previously-labelled object within the scene, this detection is discarded. We also use a technique called raycast to obtain the position where the AR label is displayed. Raycast, depicted in Figure 1, is a rendering technique to create a 3D perspective in a 2D map. The iOS app creates a raycast query by using a 2D screen location and default vector that casts outward in the z-direction from the user.

In Figure 1(a), The iOS ARKit creates a ray in the z-direction and the raycast query checks if a target exists along the ray and return a three-dimensional position of the target if so. The user can put a AR label on the detected surface. From the SpriteKit package, we can build the layout of the scene content. We use (i) SKShapeNode to build a rectangle and SKLabelNode to render the text on it, (ii) SKScene and build it to organize all active SpriteKit content, and add all nodes to the SKScene, (iii) SCNPlane to build a plane and regard SKScene as its material. When we finally build a SCNNode and set its geometry as plane, this process yields the AR labels shown in Figure 1(b).

3 A SYSTEM FOR DATA COLLECTION

Our system (see Figure 2) to reduce volunteer workload for data collection in environments like the Millar Library, comprises an iOS application for AR-based annotation (Section 2) with an Android application for indoor positioning built upon the IndoorAtlas SDK, whose data streams are integrated using a post-facto database merge. A single user

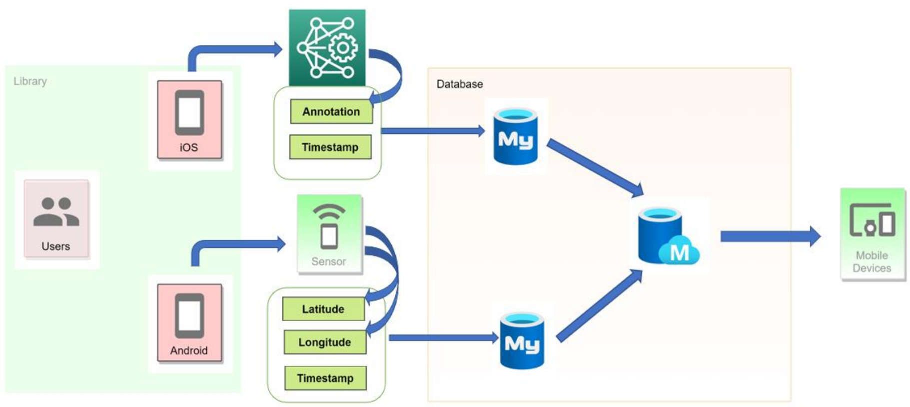
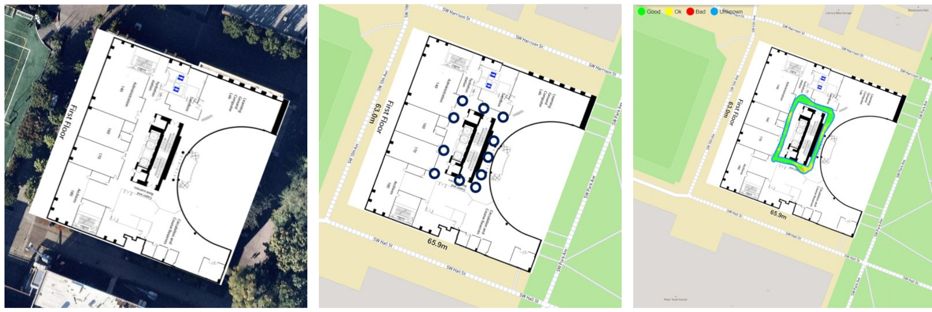


Fig. 2. System Architecture for Annotated Data Collection. Data from the AR annotation application and an Indoor Positioning Application are collected concurrently but separately and merged post-facto into a database.

(e.g. a mapping volunteer) simultaneously holds both an iOS and an Android device and slowly walks in the library. When the iOS application detects an object, the user uploads the annotation with timestamp to a MySQL database. Concurrently, the user uploads the position coordinates and timestamp of the object label, obtained from Android (Section 3.1) to another MySQL database. Both databases are merged with a join on the adjacent timestamp field. Thus, a single volunteer can replace a volunteer team for data collection tasks for applications like Nearby Explorer. In future work, we plan to eliminate the software-compatibility induced need for a volunteer to carry two mobile devices.

3.1 IndoorAtlas Based Mobile Application

Figure 3 depicts the process used to build the IndoorAtlas based mobile application. We create an IndoorAtlas location and add floor plans to it (Figure 3(a)), providing the basis for collecting mapping data on site. Our floor plan is the first



(a) (b) (c)

Fig. 3. IndoorAtlas Setup (a) Floor plan of Millar Library, (b) Waypoints added for mapping phase shown as blue dots, (c) Mapping calibrated with MapCreator2. Almost green route indicates very accurate mapping.

floor of Millar Library. We add 10 way points to pinpoint the location on the floor plan for the mapping phase with MapCreator2, an IndoorAtlas app (Figure 3(b)). Finally, we calibrate the device mapping and then map with MapCreator2 (Figure 3(c)). Now we can build our own SDK in our Android application with a license to enable location-based features.

timestamp	label	longitude	latitude
2021-04-21 13:17:45	Door	-122.68612964297652	45.51166920416415
2021-04-21 13:17:56	Painting	-122.68616661321774	45.51164277789787
2021-04-21 13:18:05	Door	-122.68619701269681	45.51162096850745
2021-04-21 13:18:13	Door	-122.68620645252369	45.51161603181833
2021-04-21 13:18:35	Elevator	-122.68623742036978	45.51154663513122
2021-04-21 13:18:42	Painting	-122.68623742036978	45.51154663513122
2021-04-21 13:18:52	Desk	-122.6862395044393	45.51152739433396
2021-04-21 13:19:01	Painting	-122.68624961321976	45.5115133736426
2021-04-21 13:19:28	Door	-122.68619706328371	45.5116248084253
2021-04-21 13:19:47	Desk	-122.68598740184828	45.511625194616556
2021-04-21 13:20:10	Desk	-122.68598231311643	45.51162595464513
2021-04-21 13:20:25	Desk	-122.68598231311643	45.51162595464513
2021-04-21 13:20:46	Desk	-122.68607698024906	45.511664898795864
2021-04-21 13:20:55	Chair	-122.6860723195348	45.51166339167316

Fig. 4. Merged database from volunteer uploaded data, with examples of annotated objects.

4 EXPERIMENT

We performed experiments in PSU's Millar Library to test our proof of concept. The iOS (version 14.4.2) and Android (version 11) devices were connected to WiFi access points in our school. We followed the 1st floor plan and detected objects, obtained corresponding location and uploaded to two databases separately. Finally, we merged the two databases into one database post facto using a join on the adjacent timestamp, as shown in Figure 4.

5 CONCLUSION

We have proposed a novel automated AR-based annotation tool for assisted indoor navigation systems like NearbyExplorer that aid people with visual impairments. Building on YOLOv3 object detection and ARKit, the AR annotation by itself, can help vision-impaired people see objects clearly and avoid obstacles. In combination with the IndoorAtlas based application, it provides an efficient system to reduce volunteer workload for data collection in environments like Millar Library. To the best of our knowledge, this is the first system to combine advances in object detection, augmented reality and indoor positioning systems for this specific purpose. In future work, we are pursuing improving the accuracy of our detection, making indoor and outdoor detection seamless. With the easing of COVID-19 restrictions, we expect to perform more comprehensive testing and evaluation of our system in diverse environments.

REFERENCES

[1] AOA (American Optometric Association. Common types of low vision. 2015.

[2] Yuhang Zhao, Sarit Szpiro, and Shiri Azenkot. Forsee: A customizable head-mounted vision enhancement system for people with low vision. pages 239–249. International ACM SIGACESS Conference on Computers and Accessibility, 2015.

[3] Yuhang Zhao, Sarit Szpiro, Lei Shi, and Shiri Azenkot. Designing and evaluating a customizable head-mounted vision enhancement system for people with low vision. ACM Transactions on Accessible Computing, Dec 2019.

[4] Yuhang Zhao, Sarit Szpiro, Jonathan Knighten, and Shiri Azenkot. Cuesee: Exploring visual cues for people with low vision to facilitate a visual search task. pages 73–84. ACM UbiComp, Sep 2016.

[5] Yuhang Zhao, Michele Hu, Shafeka Hashash, and Shiri Azenkot. Understanding low vision people's visual perception on commercial augmented reality glasses. Number 4170-4181. ACM CHI, 2017.

[6] Yuhang Zhao, Elizabeth Kupferstein, Brenda V. Castro, Steven Feiner, and Shiri Azenkot. Designing ar visualizations to facilitate stair navigation for people with low vision. UIST 2019, Aug 2019.

[7] Yuhang Zhao, Elizabeth Kupferstein, Hathaitorn Rojnirun, Leah Findlater, and Shiri Azenkot. The effectiveness of visual and audio wayfinding guidance on smartglasses for people with low vision. CHI, 2020.

[8] Dragan Ahmetovic, Cole Gleason, Kris M. Kitani, Hironobu Takagi, and Chieko Asakawa. Navcog: Turn-by-turn smartphone navigation assistant for people with visual impairments or blindness. In *Proceedings of the 13th International Web for All Conference*, W4A '16, New York, NY, USA, 2016. Association for Computing Machinery.

[9] Manaswi Saha, Alexander J. Fiannaca, Melanie Kneisel, Edward Cutrell, and Meredith Ringel Morris. Closing the gap: Designing for the last-few-meters wayfinding problem for people with visual impairments. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19, page 222–235, New York, NY, USA, 2019. Association for Computing Machinery.

[10] Brown-Ogilvie Tara Beresheim-Kools Jenna Parker Amy T., Swobodzinski Martin. The use of wayfinding apps by deafblind travelers in an urban environment: Insights from focus groups. *Frontiers in Education*.

[11] Martin Swobodzinski and Amy T. Parker. Electronic wayfinding for visually impaired travelers: Limitations and opportunities, August 2019. [Online; retrieved 17-June-2021].

[12] Shelly Brisbin. The nearby explorer blindness-focused navigation app from aph comes to ios, November 2016. [Online; retrieved 17-June-2021].

[13] Manaswi Saha, Michael Saugstad, Hanuma Teja Maddali, Aileen Zeng, Ryan Holland, Steven Bower, Aditya Dash, Sage Chen, Anthony Li, Kotaro Hara, and Jon Froehlich. *Project Sidewalk: A Web-Based Crowdsourcing Tool for Collecting Sidewalk Accessibility Data At Scale*, page 1–14. Association for Computing Machinery, New York, NY, USA, 2019.

[14] Cole Gleason, Anhong Guo, Gierad Laput, Kris Kitani, and Jeffrey P. Bigham. Vizmap: Accessible visual information through crowdsourced map reconstruction. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '16, page 273–274, New York, NY, USA, 2016. Association for Computing Machinery.

[15] Dragan Ahmetovic, Daisuke Sato, Uran Oh, Tatsuya Ishihara, Kris Kitani, and Chieko Asakawa. Recog: Supporting blind people in recognizing personal objects. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery.

[16] Apple Inc. https://developer.apple.com/documentation/arkit. 2017.

[17] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. CVPR, 2018.