ISSN No:-2456-2165

Using Computer Vision and Augmented Reality to Aid Visually Impaired People in Indoor Navigation

Akshay Anand
Cupertino High School
Cupertino, USA

Abstract:- Navigating indoor environments can be challenging for visually impaired people, particularly for wayfinding tasks. With tools like GPS, outdoor navigation is more feasible, however, when indoors, receiving low-precision location data and avoiding obscure obstacles pose a challenge. We propose an app that combines state-of-the-art advances in promptable image segmentation from computer vision and augmented reality to assist the visually impaired in indoor navigation. Due to a broader range of objects indoors, automatically detecting obstacles in real-time is challenging. The key idea in our approach is to use a faster variation of Meta's Segment Anything Model (FastSAM) to segment objects in the user's path. We use a generic indoor map of the environment to localize the user's position and overlay AR arrows that guide their navigation. FastSAM's zero-shot recognition capabilities allow us to automatically add nearby obstacles in real-time to the indoor map so the wayfinding can be updated to avoid these. Although FastSAM's speed enables our app to be deployable in real-time, the performance tradeoff from the original model makes mask generation less precise. Overall, our app can detect larger obstacles, such as chairs and tables, at a high rate and generates optimal paths to reach a destination. Many existing indoor navigation systems highly depend on a detailed indoor map or an extensive 3D environment model and don't account for dynamic obstacles. This system minimizes the amount of initial data needed and can account for obstacles that cannot be observed from a map.

I. INTRODUCTION

In the US, over 10 million adults grapple with visual impairments, including blurred vision, near or far-sightedness, or complete blindness. Witnessing my grandfather's daily challenges due to age-related macular degeneration has motivated me to find a solution for indoor navigation difficulties faced by the visually impaired. Despite significant advancements in GPS technology for outdoor navigation, indoor usage faces signal interference issues due to walls and structures. While solutions for this such as Bluetooth and Wi-Fi beacons have emerged, they often prove inaccurate in areas with weak connectivity. We define this problem of indoor navigation for the visually impaired as the following: guiding the user in the path to their destination while averting them from unexpected obstacles on the way.

Aiming to address this problem, our research builds upon the work of Dr. Nirupama Bulusu and Pei Du from their publication "An Automated AR-Based Annotation Tool for Indoor Navigation for Visually Impaired People." They utilize object detection ML models and augmented reality (AR) to allow volunteers to label environments with AR markers that will provide visually impaired individuals with spatial awareness. After discussions with them, I've identified three critical areas for improvement in their tool: detecting obstacles that the ML model is untrained on, reducing the need for extensive prior labeling, and enhancing the tracking of the user's position. Our new app aims to address these areas to create a more robust indoor navigation tool for the visually impaired.

II. METHODS

Our app utilizes the ARKit framework and Unity's AR Foundation to create an immersive scene with AR objects, such as arrows and caution symbols. Using computer vision object detection and ray casting, we identify the real-world position of nearby obstacles and place AR caution symbols there to alert the user. Our computer vision system utilizes Meta's Segment Anything Model (SAM) for zero-shot image segmentation. SAM, a more effective alternative to the YOLO object detection model, excels in adapting to unseen objects that are more prone in cluttered indoor settings. We implement FastSAM, a streamlined variant with faster inference speeds and minimal performance tradeoffs.

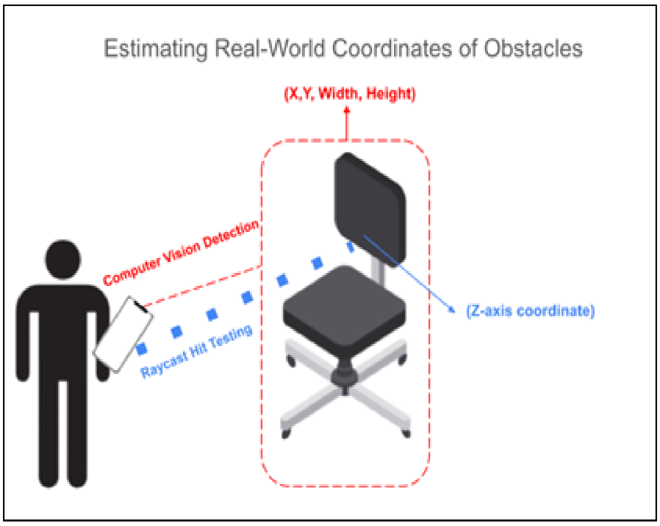
Estimating Real-World Coordinates of Obstacles
(X,Y, Width, Height)
Computer Vision Detection
Raycast Hit Testing
(Z-axis coordinate)

Fig 1 System for 3-D Coordinate Estimation of Indoor Objects

We use a floor plan diagram, a 2D grid containing filled squares (walls or blockages), and empty squares (navigable areas), to represent the indoor environment. Our navigation task involves estimating the position of the user on this floor plan diagram so that our app can find the path to our destination. The phone camera scans nearby visual markers, in this case, QR codes, and localizes us on the 2D floor plan, updating it in real-time with the user's movements. For navigation, we employ the A-star path search algorithm to determine an optimal trajectory to the user's desired destination. We place AR arrows in the scene that guide the user on this calculated path.

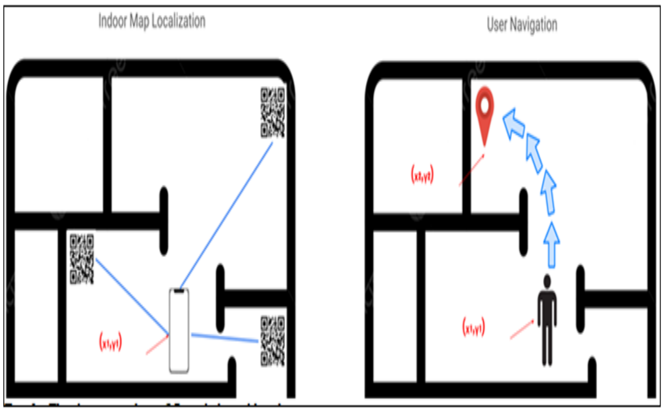

Fig 2 The Diagram of our QR-Code based Localization System

III. RESULTS

In selecting our computer vision model, we considered three options: SAM, FastSAM, and MobileSAM. To determine the optimal choice, we evaluated their speed and precision. When comparing speed, FastSAM and MobileSAM outperformed SAM, with runtimes of approximately 2 seconds compared to SAM's 8 seconds. To assess segmentation mask precision, we evaluated these models on 15 indoor object images from the MITADE20K dataset using the IOU metric *(accuracy of generated masks).* SAM achieved the highest IOU, just below 0.8, followed closely by FastSAM with an IOU of around 0.75. MobileSAM had a significantly lower IOU, indicating a tradeoff between speed and performance. Ultimately, we selected FastSAM for its balanced accuracy and speed.

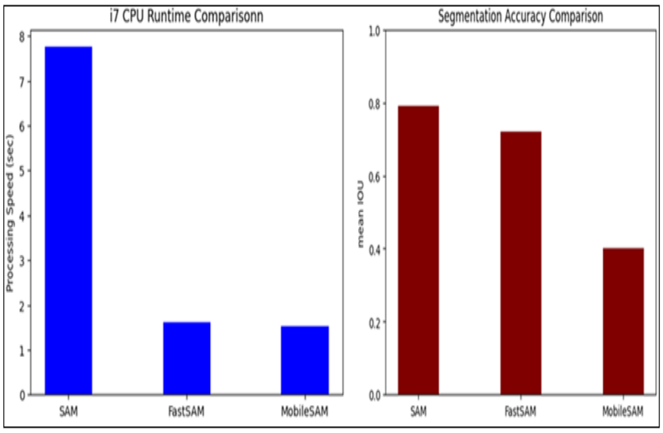

Fig 3 Performance and Speed Comparisons of SAM Model

Below are three examples of our model's performance on obstacles in our indoor test environment, highlighting its strengths and weaknesses. FastSAM nearly perfectly segments the chair (right) and the fan (left). It does a good job of locating obtruding things in the camera's view that easily stand out. However, in the middle, it's evident that our model should be able to segment the couch footstool, but instead segments a small portion of the couch behind it. FastSAM tends to undergeneralize when generating segmentation masks, by highlighting specific portions of one object rather than identifying the entirety of it. For obstacle avoidance, our model is able to calculate the pixel coordinates of the center of the object, which we transform into real-world X and Y coordinates.

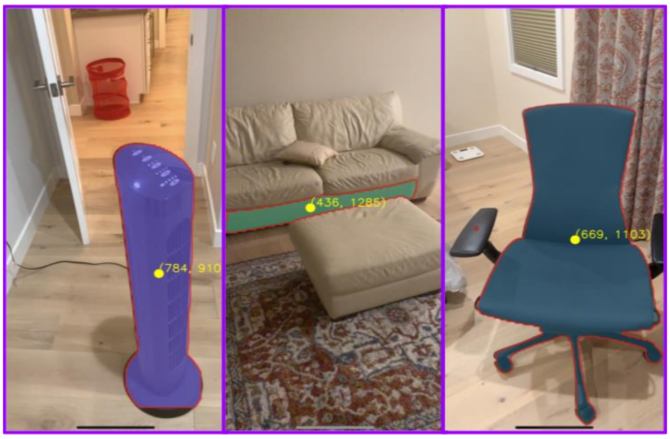

Fig 4 2-D Coordinate Estimation with SAM Masks

After pinpointing obstacles on-screen through computer vision, we use ray casting to determine their real-world 3D coordinates and position an AR caution symbol accordingly. In the accurate example on the right, the caution symbol is placed perfectly on the chair. However, in the left example with the fan, the AR symbol's left tip is against the wall, instead of being on the fan in front of it. This inconsistency in ray cast hit-testing is evident; it tends to hit flatter surfaces like the chair and wall and passes through solid objects like the fan.

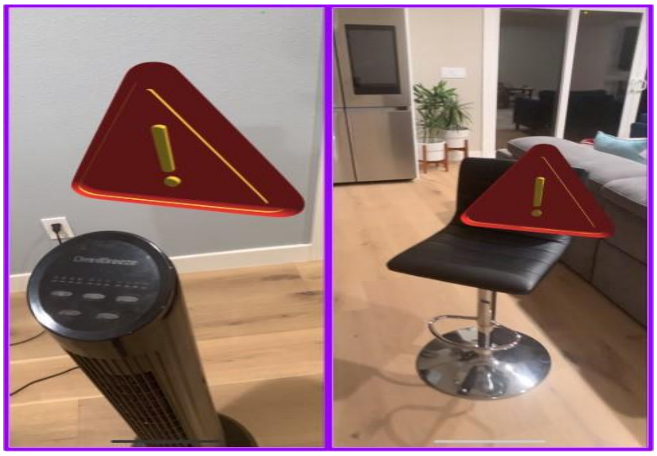

Fig 5 AR Caution Indicator Placement

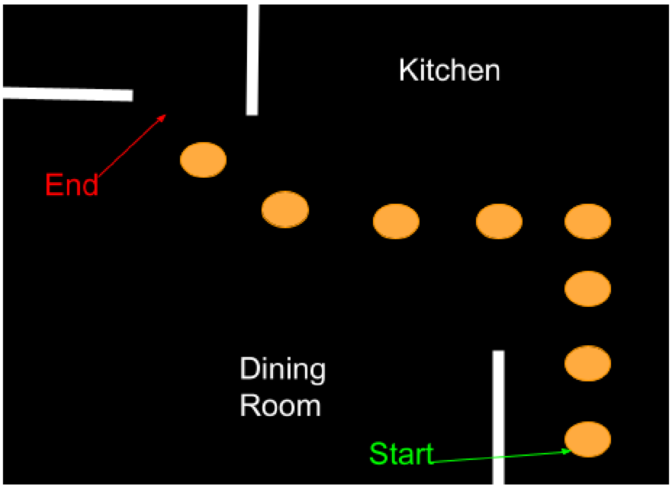

Fig 6a Mapping and Pathfinding of our Approach

To conserve memory, we use AR spheres instead of AR arrows for navigation. The map illustrates our task of moving from the hallway (labeled 'start') to the bedroom (labeled 'end'). Our app tracks our position on this 2D map through localization and determines an optimal path to our destination, guiding us with AR spheres. The spheres lead us between the dining room and the kitchen and towards the bedroom doorway. When traversing longer paths, however, our app may struggle due to accumulated inaccuracies in our 2D position estimation, leading to the misplacement of AR spheres. Nevertheless, for shorter paths, our navigation guidance remains accurate.

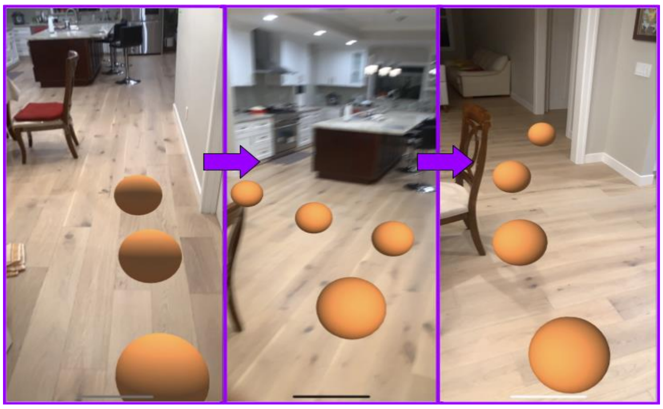

Fig 6b AR Sphere-based Path Guidance

The graph below portrays how well our localization system works. As expected, the more we move around while the app is open, the more the error of our 2-D estimated position. In our test environment, there was 1 QR code in each room, so the app didn't always relocalize between each interval of movements. Still, our error stays relatively low at around 4 meters maximum per axis even after moving 35 meters. We can see that overall, the trend between the error of the estimated x-coordinate in the 2-D plane is fairly consistent with that of the y-coordinate, both steadily increasing.

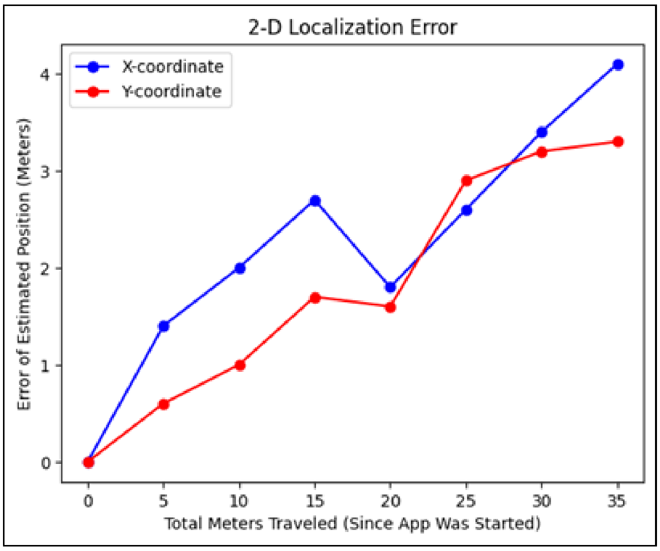

Fig 7 Map Localization Error in Meters

IV. DISCUSSION/CONCLUSION

Our app provides essential functionalities for visually impaired users, enhancing their indoor navigation safety. For straightforward routes, we display distinct AR spheres that guide users to their destination. Our computer vision model excels at obstacle detection, placing AR caution symbols when necessary. However, app speed is currently a challenge, as most computations occur on a CPU server and then return to the phone. Because our slow runtime, we can only detect one obstacle at a time, causing potential danger when multiple obstacles exist. Transitioning to a T4 GPU server could dramatically boost processing speed, making our tool more reliable.

While our segmentation is generally accurate, there are instances where it identifies parts of obstacles, rather than the entire object. To address this, fine-tuning FastSAM on the ADE20k dataset may enhance its recognition of indoor objects. Our model outperforms YOLO by detecting a broader range of objects, reducing false negatives, and improving user safety. Using ray casting for calculating 3D positions of obstacles is unreliable due to its tendency to pass through surfaces. LiDAR sensors could be a more dependable alternative, though not all devices support them. In terms of navigation, the QR code-based localization method is inconvenient, requiring users to get close to them for accurate positioning. This limitation restricts navigation to relatively short distances.

Our app showcases alternative computer vision methodologies. Unlike traditional use cases of zero-shot models, which segment images into regions of interest, our app uses FastSAM to process these regions and identify objects of interest. This enables object detection. The adaptation of segmentation models for detection purposes highlights the potential of FastSAM's self-supervised learning ability to solve a wide array of ML problems. Many indoor navigation technologies rely solely on AR arrow-based guidance. In contrast, our app combines this with real-time obstacle detection using computer vision, providing navigation capabilities while also enhancing safety for visually impaired users.

REFERENCES

[1]. 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. https://doi.org/10.1145/3441852.3476561

[2]. Xu Zhao, , Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, Jinqiao Wang. "Fast Segment Anything." (2023).

[3]. Chaoning Zhang, , Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, Choong Seon Hong. "Faster Segment Anything: Towards Lightweight SAM for Mobile Applications." (2023).

[4]. Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, Ross Girshick. "Segment Anything." (2023).

[5]. Zhang, Huijuan, Chengning, Zhang, Wei, Yang, Chin, Chen. "Localization and navigation using QR code for mobile robot in indoor environment.". 2015.

[6]. Watthanasak Jeamwatthanachai, , Michael Wald, Gary Wills. "Map data representation for indoor navigation by blind people". *International Journal of Chaotic Computing* 4. 1(2017): 70–78.

[7]. Maria deFatima X.M.Almeida, , Laura B. Martins, Francisco J. Lima. "Analysis of Wayfinding Strategies of Blind People Using Tactile Maps". *Procedia Manufacturing* 3. (2015): 6020-6027.

[8]. B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso and A. Torralba, "Scene Parsing through ADE20K Dataset," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 5122-5130, doi: 10.1109/CVPR.2017.544.

[9]. Daniel Foead, , Alifio Ghifari, Marchel Budi Kusuma, Novita Hanafiah, Eric Gunawan. "A Systematic Literature Review of A* Pathfinding". *Procedia Computer Science* 179. (2021): 507-514.

[10]. R. C. DuToit, J. A. Hesch, E. D. Nerurkar and S. I. Roumeliotis, "Consistent map-based 3D localization on mobile devices," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 6253-6260, doi: 10.1109/ICRA.2017.7989741.

[11]. Yilong Zhu, Bohuan Xue, Linwei Zheng, Huaiyang Huang, Ming Liu, Rui Fan. "Real-Time, Environmentally-Robust 3D LiDAR Localization."(2019).

[12]. H. Kim, T. Oh, D. Lee and H. Myung, "Image-based localization using prior map database and Monte Carlo Localization," 2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Kuala Lumpur, Malaysia, 2014, pp. 308-310, doi: 10.1109/URAI.2014.7057440.