Navigation and Interaction for Blind Users via a Cognitive Architecture

Oscar J. Romero, Anthony Tomasic, Elizabeth J. Carter, John Zimmerman, Aaron Steinfeld

Carnegie Mellon University
oscarr@andrew.cmu.edu, tomasic@cs.cmu.edu, ejcarter@andrew.cmu.edu, johnz@andrew.cmu.edu, steinfeld@cmu.edu

Abstract

Navigating new indoor spaces and interacting with the environment presents many challenges for people who are blind or low-vision (BLV). To address these challenges, we prototyped a smartphone-based conversational assistant that helps BLV people navigate and interact with their environment. The prototype utilizes a cognitive architecture to integrate three different technologies: (i) *augmented-reality spatial anchors* for high-precision localization and access to static information about the environment; (ii) *real-time object/people detection* for information about the environment and obstacle avoidance; and (iii) a *conversational agent* that uses large language models (LLMs) for information extraction, conversational interaction, and turn-by-turn navigation. We assess the impact of different technologies on human performance by measuring user task time and errors. We found that conversational interaction holistically integrates the different technologies to deliver a better user experience while significantly reducing task completion time.

Dataset — https://github.com/CMU-TBD/ar-od-llm-indoor-navigation

Introduction

Navigating unfamiliar indoor spaces and interacting with objects and situations in the environment present a significant challenge for people who are blind or low-vision (BLV). Attending a conference requires people to navigate the venue, find the desired room, and discover and approach a microphone to ask a question. A collection of emerging technologies provides new solutions to this problem: augmented reality (AR), high-precision indoor localization, large language models (LLMs) that support conversational systems, and computer vision for scene understanding and real-time object detection and identification. AR (Microsoft 2023a; Lan 2024; Guerreiro et al. 2019; Schieber et al. 2024) provides a new channel of information by placing markers in the environment (spatial anchors), allowing for navigation and access to static information about the surroundings. This functionality makes it possible to navigate and partially understand context. Finally, LLMs open the door for effective conversational interaction, allowing people to engage with a navigation system, inquire about their environment, and learn about things that they might want to interact with. The ability to talk to an agent that understands navigation and interaction (Aira 2023; Eyes 2023; Microsoft 2023a; Lan 2024; Sato et al. 2019b) affords a new type of user experience, opening up the world for more independence and quality of life for BLV people (OrCam 2023; Aira 2023; Eyes 2023; Microsoft 2023a; Sato et al. 2019b).

In this paper, we study the impact, individually and collectively, of these new technologies on the design of navigation and interaction assistants for BLV people. We utilized co-design to create a prototype that utilizes a novel combination of technologies to provide a better solution. Thus, our research focuses on the following research questions.

1. What is an effective integration of object detection, AR spatial anchors, and conversation in support of BLV people's navigation and interaction in the world?
2. What task performance improvement is provided by integrating conversational interaction, object detection, and AR spatial anchors?

To evaluate technology impact, we created four tasks derived from our co-design process, each comprising interaction subtasks and operations. Through a study, we measured the time for each operation, breaking down the total task time. These results pinpoint specific areas for further research. Thus, our contributions are:

1. A prototype that demonstrates the impact of integrating traditionally separate accessibility research areas – indoor navigation and object interaction,
2. An assessment of the improvement of human task performance as a result of integrating three technologies: conversational interaction, object detection, and spatial anchors; and
3. A description of a cognitive architecture that effectively orchestrates the integration of object detection, spatial anchors, and conversation to support both independent navigation and interaction;
4. A discussion of applications and challenges.

Architecture

Our system uses a client/server architecture (Figure 1). The system has a cognitive architecture capable of managing un-

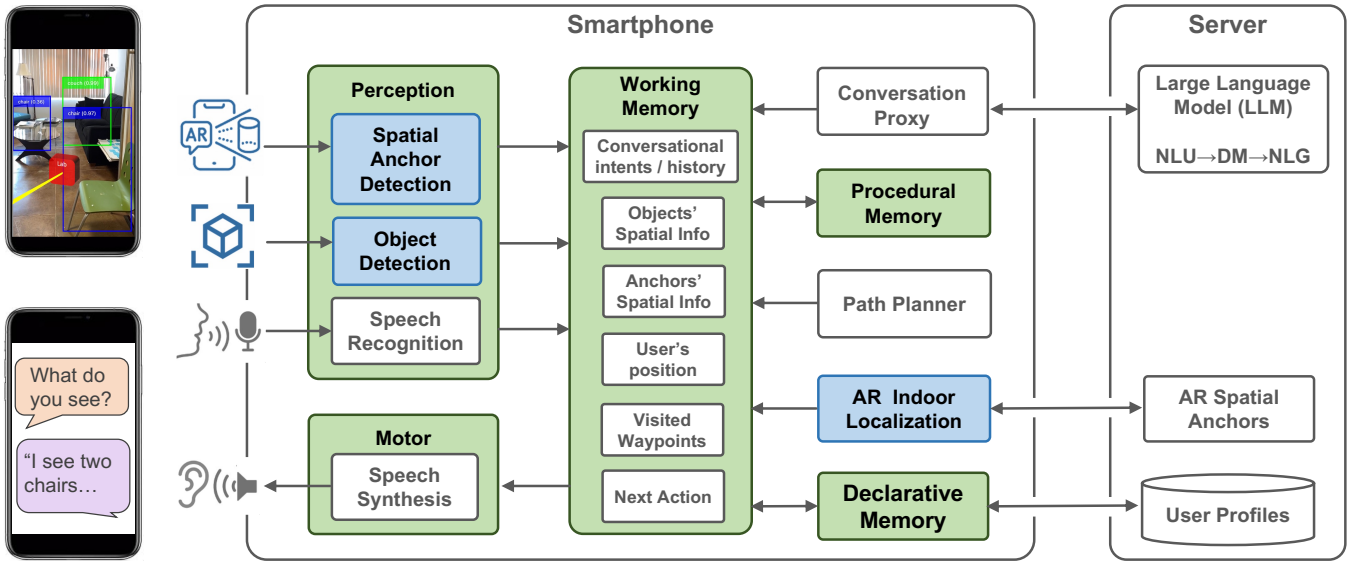

Figure 1: Modular client-server architecture. On the client side (smartphone), boxes represent modules running asynchronously on separate threads. The working memory component integrates the information from other modules and produces a speech output. A cognitive cycle initiates when the perception module processes internal and external information stored in the working memory. The procedural memory decides what to do next by retrieving the contents of the working memory, which in turn retrieves knowledge about the user and the world. Finally, the cycle ends with a conversational action processed by the motor module. Blue-colored modules can be turned on/off according to the assessment setting. Green-colored modules denote cognitive modules proposed by the CMC. Arrows indicate the flow of information.

certainty under time constraints. A cognitive architecture is a hypothesis about the fixed structures that provide a mind and how they work together to yield intelligent behavior in a diversity of complex environments (Laird, Lebiere, and Rosenbloom 2017).

The rationale for choosing a cognitive architecture approach is two-fold. (i) It prioritizes bounded rationality over optimality, enabling the system to compensate for limited resources by selectively filtering and processing continuous streams of data to make timely decisions. This feature is a critical factor considering that the app operates on limited phone resources. (ii) The interaction among the cognitive components facilitates emergent control and orchestration of object detection, spatial anchors, and conversation.

Overall, our system is inspired by the Common Model of Cognition (CMC) (Laird, Lebiere, and Rosenbloom 2017). CMC proposes a reference cognitive architecture that defines five high-level modules, including perception, motor, working memory, declarative long-term memory, and procedural long-term memory. We integrate the modules proposed by the CMC along with an LLM, following a *modular approach* where the cognitive architecture augments an LLM by injecting reasoning traces and contents from memories into the prompting process ((Romero et al. 2023)).

Perception Module

The Perception module yields symbol structures (percepts) with associated metadata in specific working memory buffers. The system perceives the world through three components: speech recognition, object detection, and spatial anchor detection. Speech recognition perceives spoken words and emits a text transcription. Object detection performs real-time multi-object detection and tracking by using the live camera feed. It uses mobile app-optimized, off-the-shelf APIs for computer vision and leverages a pre-trained image classification model that recognizes object labels and categories, such as food, furniture, home/office appliances, building parts, and people.

For each detected object, object detection returns a set of labels, a confidence score per label, and a bounding box representing the location of the object on the screen. Spatial anchor detection recognizes AR spatial anchors, that is, 3D models that represent points of interest that can have attached virtual content. Spatial anchors rely on an AR framework to perceive the environment and track the device's movement, position, and orientation. The AR framework uses visual-inertial odometry, a technique that combines information from the device's motion-sensing hardware (e.g., compass, accelerometer, LiDAR scanner) with computer vision analysis of the scene visible to the device's camera (Example 1).

Working Memory

The conversational navigational guidance and interaction with objects emerge from the interplay of all the modules through the working memory, which provides global communication. The working memory locally stores the situational context of the interaction into specialized buffers, for instance, the current navigational route and waypoint, the conversational history, objects' and anchors' spatial in-

Example 1: Example of perception outputs.

```
(perceives, speech_recognition,
     [Where is the fridge?])
(perceives, object_detection,
     [(microwave, pos1)...])
(perceives, spatial_anchors,
     [(fridge, pos2), ...])
```

Example 2: Example of 12 special tokens LLM input.

```
<bos><user>do you see a microwave?
<history>user: hi. sys: hi, how can I help
you?
<objects>(table, left), (microwave, in-front)
<distance>far-from-waypoint
<angle>30 degrees
<planner>turn-slightly-right
```

```
<intent>search-object
<entity>microwave
<call>is_perceived(microwave): bool
<output>Yes, I see a microwave in front
of you <eos>
```

formation, the user's current position, already-visited waypoints, and past and current actions. The contents of the working memory decay in proportion to their salience.

Conversational Proxy and Large Language Models

The conversational proxy, in combination with an LLM, interprets the intentions that the user conveys and provides assistance conversationally. The proxy structures the contents of the working memory in a specific format and prompts the LLM to generate an output that is then stored back in the working memory. To this end, we fine-tuned an LLM and encoded a set of 12 special tokens, which serve as delimiters and segment indicators (Example 2).

Special tokens <bos> and <eos> indicate the beginning and end of the sequence, respectively. <user> denotes the most recent user's utterance, <history> contains the most relevant conversational history stored in the working memory, <objects> is a list of perceived objects and anchors and their relative position in the real world, and <distance> and <angle> correspond to the current orientation of the user with respect to the next waypoint anchor in the path. <planner> is the next navigational step generated by the path planner. <intent> and <entity> are the dialogue intent and entities extracted from the user's utterance, respectively. <call> is an internal invocation to retrieve content from a specific buffer in the working memory. <output> is either a navigational instruction or a system's response to a user's request. When prompting the LLM, the subsequence <bos>...<intent> (above the line) is used as an input, so the LLM generates the rest of the sequence, providing an intent,a list of entities, a call (if any), and a speech output (below the line).

We fine-tuned an LLM that generates the inputs/outputs of three main components in a manner that resembles the execution of a pipelined spoken dialog system (Chen et al. 2017; Romero et al. 2021). The execution pipeline starts with a Natural Language Understanding (NLU) component that leverages the LLM model to process the prompt generated by the conversational proxy and classify a set of both user intents and entities.

Then, the Dialogue Manager (DM) component controls the conversation flow, tracks user information, and builds conversational context via the working memory by performing a two-step process. First, the DM takes the outputs from the NLU component along with the conversational history and prompts the LLM model to generate both a dialogue state (containing a goal constraint, a set of slots/values, and the dialogue act) and a function call signature that maps the slots/values in the dialogue state onto arguments of a function call that retrieves specific contents from the working memory (e.g., the call `get_detected_objects()`). As a second step, the DM injects the results from the function call into the pipeline, and the LLM model is prompted again to generate a set of intents/actions that the system should perform in the current conversational turn.

Finally, a Natural Language Generator (NLG) component dynamically converts the system intents generated by the DM into natural language text. To that end, the LLM model maps the system intents onto natural language templates, and then it replaces text placeholders with information stored in the working memory. The generated text is then synthesized by the speech synthesis module.

As a backup feature, when the smartphone cannot communicate with the server due to internet connection issues (e.g., inside an elevator), the conversational proxy uses on-device NLP libraries for sentence embeddings and entity recognition to locally recognize the user's intents until the connection with the LLM is reestablished.

Procedural Memory Module

The procedural long-term memory exerts global control by modifying the contents of working memory through the activation of rules. For instance, a set of rules is used to filter the objects perceived by the system and assign the most representative classification label. As mentioned before, each object is classified into one or multiple labels, exhibiting a wide range of detail levels, with higher confidence assigned to the more general labels.

Additionally, a set of procedural rules prescribes the execution priority of the system's audio and speech output and decides whether a process must be interrupted to make way for another one. For instance, the non-verbal audio feedback that signals the user when they need to reorient their position has the lowest priority. The next priority level corresponds to the verbal turn-by-turn navigational instructions. Next, the mechanism that verbally lists all the detected objects when requested by the user has a higher priority level. Finally, answers to the user's questions have the highest priority. Another subset of rules is in charge of pausing speech synthesis if speech recognition is running simultaneously. Speech synthesis is resumed after speech recognition is done. As a result, the motor module adds system actions to a queue so they are executed in a particular order.

Declarative Memory Module

The declarative memory is separated into semantic and episodic memories. The former maps to semantically abstract facts, while the latter maps to contextualized experiential knowledge. Semantic memory stores both the user's navigational preferences (e.g., speech rate and volume, distance units, etc.) and each spatial anchor's virtual content as semantic chunks, for instance: `(chunk1, door, (material: glass, opens: push, handle: panic exit bar))`. The episodic memory stores past user interactions with the system to enhance the personalization of navigation for previously visited places. This aspect of the work is currently in progress.

Indoor Localization and Path Planner

Using the sensors' inputs, the AR framework infers and generates a 3D sparse point-cloud representation of the world, and, by matching the point-cloud map against the feature points generated in real-time by the camera, it provides a way for the device to recognize spaces and precisely localize its position and orientation within that space, with accuracy typically at the centimeter scale.

Once spatial anchors for waypoints are placed in the real world and persisted in the server, we build a weighted graph representing how waypoints are connected to each other, where the cost of each edge corresponds to the distance between waypoints. Next, during execution, our path planner uses the A^* path search algorithm to determine the optimal (least-cost) path for the user to follow.

The spatial anchor recognition system re-scans the surrounding space every 10 seconds to ensure proper localization of anchors. User safety is a top priority and an ongoing research problem. For example, safety considerations appear before and during route setup: high-quality spatial anchors are attached to immovable objects (a door, as opposed to a chair) and are not attached to risky objects (such as a crosswalk). Safety issues also occur during navigation: cross-checking between multiple models (for example, the spatial anchor is labeled as a curb, and the real-time object detection confirms the curb). Safety design is part of the conversational system: when evidence is contradictory, the user can be notified that the system is uncertain. Any inconsistencies are marked for re-examination.

Motor Module

The Motor module converts symbolic relational structures in the working memory into speech synthesis actions. Speech synthesis sets two different speech rates: normal (1.0x) for conversational interaction, such as answering user questions, and fast (2.0x) for providing navigational instructions, like "walk 5 steps forward". This design aligns with the common practice in speech synthesis for BLV people, who adapt to accelerated speech in scenarios where the discourse involves a familiar set of sentences with minor variations, facilitating easy recognition (Branham and Mukkath Roy 2019). In the case that the speech is novel, a lower speed is used to ensure comprehension of the audio (Example 3).

Example 3: Example of a motor action.

```
(action: utter, (
   (text, "The microwave is in front of you")
   (type, question_answering)
   (speech_rate, 1.0) ))
```

System Operation

The modules of the cognitive architecture work together to make timely decisions based on perceived information, context, and user requests. This interaction among modules allows the system to operate in two modes: navigation and interaction. In navigation mode, it handles route planning, wayfinding, and context-aware notifications. For example, the system can inform users about nearby restrooms, warn them of obstructions, or notify them if someone is in front of them. These notifications cover both static and dynamic aspects of the environment. For static elements (e.g., a restroom), the system retrieves virtual content stored in AR spatial anchors, while for dynamic elements (e.g., an approaching person), it relies on real-time object detection.

In interaction mode, the system offers various capabilities. It helps users locate objects through object detection (e.g., finding keys), and users can ask questions about how to interact with specific objects and their state (e.g., how to open different types of doors, is the door open). The system also supports multi-turn conversations about specific locations (e.g., What is inside the cabinets?) and allows users to inquire about their surroundings, requiring scene understanding (e.g., how many people are in the room). To support these capabilities reliably, particularly in real-world conditions where network latency and model coordination can pose challenges, the system uses several technical strategies.

The system mitigates latency and consistency issues in multi-LLM integration through a combination of architectural, procedural, and fallback strategies, inspired by the Common Model of Cognition. Lightweight models like GPT-2 handle fast-turnaround tasks such as intent recognition, while heavier models are reserved for deeper reasoning and multimodal understanding. Inconsistent outputs between models are resolved through a centralized working memory and a set of procedural control rules that synchronize updates and prioritize which information is acted upon. When network delays or connectivity losses occur, the system maintains functionality (as much as possible) using on-device NLP fallbacks. To prevent prompt desynchronization, the conversational proxy formats all inputs and outputs using a consistent schema that encodes a shared context across models. A latency-aware dialogue scheduling ensures that navigation instructions are delivered in sync with user movement, preventing confusing overlaps or delays in speech.

Assessment

We designed our assessments to study the impact of each functional subsystem. This design informs both usability research and technical decisions. The assessment design involves a realistic navigational and interaction environment within a real-world building. The factors of the assessment

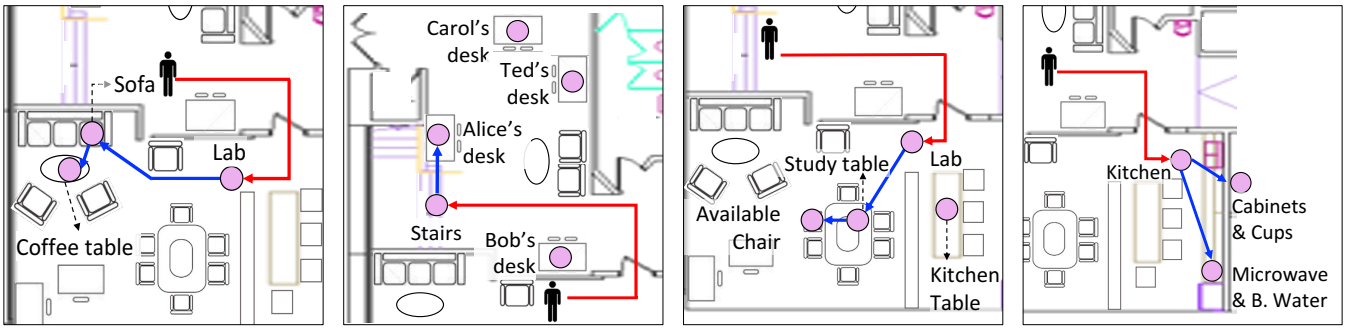

Figure 2: Floor plan for the four assessment tasks. a) Task 1: Navigate from the corridor to the sofa and find a mineral water bottle. b) Task 2: Navigate from the lab to the stairs and find the laptop on top of the closest desk. c) Task 3: Navigate from the corridor to the study table and find an available seat. d) Task 4: Navigate from the corridor to the kitchen, find a cup inside the cabinets, and find a water bottle. Pink circles represent spatial anchors. Red arrows represent navigation routes and blue arrows designate a combination of navigation and object interaction.

include two technology choices: object detection and AR spatial anchors. These factors were selected to analyze their impact on task performance. Task completion time is measured to capture a central benefit of the system. We also analyzed errors in the interactions to provide directions for future work.

Technical Details

We ran the assessment scenarios on Apple iPhone 13 Pro devices equipped with iOS 16, an A15 Bionic processor (16-core neural engine, 6-core CPU, and 5-core GPU), 256GB of memory, a TrueDepth camera, three rear cameras, and a LiDAR scanner. We chose this hardware configuration so that on-device machine learning models for object/person detection performed with high accuracy, and the localization of spatial anchors was accurately determined.

With respect to the software technologies we employed, both speech recognition and speech synthesis use the core framework delivered with iOS. The object/person detection module harnesses the power of Google's MLKit (Google 2025), an on-device framework for computer vision. We also use an image classification model pre-trained on TensorFlow (TensorFlow 2025) that can recognize up to 630 object labels and 37 categories. As frameworks for augmented reality and spatial anchors, we used Apple ARKit (ARKit 2025) and Azure Spatial Anchors (SpatialAnchors 2024), respectively.

For the conversational module, we used three different LLMs: GPT-2 (GPT2 2024) for intent and entity recognition, Mistral (Mistral 2024) for question answering, and LlaVA (LlAVA 2024) for vision and scene understanding. The use of three distinct LLMs is motivated by the specialization of each model for different tasks, leveraging their strengths and compensating for their weaknesses. GPT-2, as a smaller pre-trained model, processes prompts more quickly than Mistral. In contrast, Mistral, a significantly larger pre-trained model–approximately ten times larger than GPT-2–demonstrated greater robustness in question-answering tasks, albeit with longer response generation times. Therefore, GPT-2 was a more suitable candidate than Mistral for intent recognition and entity extraction tasks, both of which require rapid processing. Meanwhile, Mistral excelled at browsing and extracting information from spatial anchors to effectively answer users' questions. Finally, we integrated LlaVA due to its support for multimodal processing, which includes both language and vision. This capability allows the system to interpret scenes and answer questions based on images taken from the user's surroundings.

To mitigate the typical hallucination effects common in LLMs (Yao et al. 2024), we fine-tuned GPT-2 specifically for intent recognition and entity extraction. During fine-tuning, we used a training dataset comprised of 1K examples extracted from dialogue logs collected from a co-design study. To diversify the dataset, we employed data augmentation techniques utilizing LLMs, generating a range of variations based on the initial dataset. Our dataset is structured similarly to the DSTC-8 schema-guided dialogue dataset (DSTC8 2021), emphasizing intent/entity recognition and dialogue state. We fine-tuned a GPT-2 small-size model (355M) over 8 epochs using a gradient-based method, with specific hyperparameters: Adam optimizer with a learning rate of 5.75e-5, epsilon value of 1e-8, temperature of 0.5, and top-p nucleus sampling with a value of 0.95, alongside top-k sampling set to 50.

We used the vanilla versions of LlaVA (7.2B parameters) and Mistral (7B parameters), that is, no fine-tuning was performed on these two models. Additionally, we used Ollama (Ollama 2024), an open-source tool that allows running LLMs locally on a server or computer. Using Ollama will facilitate porting the LLMs from the server to the mobile device in the future.

Study and Data Collection

The study, approved by our Institutional Review Board, involved eight participants: five women and three men. Of these, six were blind, one had blindness in one eye, and one

had low vision. The ages of the participants ranged from 35 to 78, with a mean age of 61.7 years (SD = 15.8).

Our study faced challenges due to the limited population of visually impaired individuals in the city where it was conducted. Recruitment relied on both the presence and willingness of the local population to participate, which was a significant obstacle. Despite the assistance of local disability organizations and the offer of a relatively high participation incentive ($50 for a 40-minute study), the recruitment of visually impaired participants proved challenging. However, the sample size of eight participants was sufficient to (1) provide insights into the usability of such systems (according to the literature (Nielsen and Landauer 1993; Fountain 2020), only 5-10 participants are required for usability assessment), and (2) inform future functionality and design of similar systems.

Functionality Choices Our experimental design defines three functional choices that enumerate the factors to be analyzed:

- Functionality choice C1: Participants perform the task using our technology with the object detection feature enabled.
- Functionality choice C2: Participants use our technology with the AR spatial anchors feature enabled.
- Functionality choice C3: Participants use our app with both the object detection and the AR spatial anchors features enabled.

Tasks We defined four tasks, ensuring that each participant completed each one under a different functionality choice (see Figure 2).

- Task T1: The participant is tasked with navigating from the corridor to the lab, locating the sofa, taking a seat, finding a water bottle on the coffee table, and ensuring they drink the mineral water, not the sparkling one.
- Task T2: The participant is instructed to return a pen borrowed from Alice. They must walk from the lab to the stairs, find the closest desk to the stairs (Alice's desk) among the other desks, locate a laptop on the desk, and leave the pen next to it.
- Task T3: The participant has a lab meeting, requiring them to travel from the corridor to the lab, find the study table (there is also a kitchen table in the room that must be not confused with the study table), and locate an available chair to sit on.
- Task T4: The participant desires to drink water. Starting from the corridor, they are to navigate to the kitchen, search for the cabinets, find a cup, locate the microwave (as water bottles are nearby), then find the water bottle and pour its contents into the cup.

We use a balanced Latin square design to assign functionality choices per task to each user, preventing the carry-over effect. With eight participants recruited, we assigned functionality choices to tasks for the initial four participants and then replicated the same arrangement for the remaining four participants. The order of task execution was randomized to mitigate the impact of becoming familiar with the spatial layout.

Task	Functionality choices		
	C1	C2	C3
T1 (R)	240.0±25.0	245.5±99.5	149.0±40.0
T2 (R)	131.5±34.5	186.5±84.5	105.5±24.5
T3 (R)	135.0±31.0	70.5±8.5	73.0±22.0
T4 (R)	289.5±106.5	275.5±124.5	165.5±5.5

Table 1: Average total time (sec) and standard deviation for completion times per functionality choice.

Apparatus During the study session, all participants were given the option to either wear a mobile phone chest mount for a hands-free experience or hold the phone in their hands. However, some participants experienced discomfort with the chest mount due to variations in their body types, and it required them to rotate their entire torso to orient the phone properly to locate each anchor. Consequently, we opted to have participants hold the phone in their hands to facilitate its operation. Throughout the session, the app logged every event in our database, and a researcher closely monitored and videotaped all participants using an external camera.

Before performing the tasks, participants engaged in a 20-minute practice session to familiarize themselves with the use of each operator in different situations. Participants were given the same detailed instructions to perform the task regardless of functionality choice. The performance time metric begins at the moment the participant utters a command/request and continues until the operator completes the task, including any errors requiring operator repetition.

Findings

The presented results allow us to identify some trends and patterns in the use vs. non-use of our technology and the impact of having certain features of the system disabled.

Task Completion Time

We averaged the total time taken by participants to complete each task (Table 1). The fastest completion time was observed in the functionality choice where both object detection and spatial anchors were enabled (C3), except for Task 3. In this case, participants benefited from prior knowledge of the arrangement of the lab gained by completing other tasks.

Error Analysis

We identified two macro error categories: repetitions and redirections. Repetitions occur when a participant repeats a request or command due to a system error. Redirections happen when the researcher redirects a participant because they deviate from the task's steps.

We further broke down each macro category into subcategories. Repetitions mainly occurred due to mislabeling errors in speech recognition or object detection and localization errors caused by misplaced spatial anchors (i.e., accumulated errors in pose estimation). In contrast, redirections

Category	Subcategory	Avg.	Perc.
Repetitions	Speech recognition errors	2.4	48.1%
	Object detection errors	1.8	37.2%
	Mislocated spatial anchors	0.7	14.7%
Redirections	Exploring commands	1.8	47.8%
	Misinterpreting instructions	1.0	27.3%
	Mixing up commands	0.9	24.9%

Table 2: Identified types of errors.

account for the instances when a participant wanted to explore different commands instead of sticking with the instructed one, misinterpreted instructions from the researcher, or mixed up commands. For example, participant P4 continually confused the command "echolocate" with the wake word for Amazon's Echo device, which is "echo". We tally the total number of occurrences of these errors across tasks and present the average number of errors per task and the corresponding percentage for each error subcategory in Table 2.

Speech recognition errors occurred almost one-third more frequently than object detection errors and three times more often than mislocated spatial anchors. Conversely, participants' preferences for exploring system commands over using the instructed one nearly doubled the instances of misinterpreted instructions and mixing up commands.

QoE Metrics

Table 3 summarizes key aspects of the participant interviews. After conducting free-form interviews, we categorized and analyzed the responses statistically. Participants most appreciated the app's features, with 62.5% favoring echolocation and 50% expressing satisfaction with the ability to find an unoccupied seat. Additionally, 12.5% valued the capability to locate objects remotely. Participant P6 specifically noted that, unlike SeeingAI, our app surpasses the 6-foot range limitation, enabling users to find objects at any distance using AR technology that tracks long-distance spatial anchors.

Participants expressed dissatisfaction with the app's occasional inaccuracies in estimating distances (50%), which was expected when the app relied solely on object detection without spatial anchors. Additionally, navigating through narrow spaces proved to be particularly overwhelming because instructions overlapped (37.5%). Three participants reported that there was nothing in particular that they disliked about the app (37.5%).

In general, participants indicated a reluctance to use the app in familiar places like their homes (62.5%), mainly because they already navigate these spaces independently. However, three participants (37.5%) mentioned they would consider using the app to find keys or an umbrella if lost at home. In unfamiliar environments, participants expressed their willingness to use the app for various purposes such as exploring the surroundings of an airport gate during a long layover (62.5%), navigating a hospital for a doctor's appointment (37.5%), or finding preferred stores in a mall (12.5%). One participant suggested that the app could be

Question	Answers	Perc.
What are the app's features that you liked the most?	Echolocation	62.5%
	Finding an available chair	50%
	Finding objects remotely	12.5%
What are the app's features that you liked the least?	Distances were not accurate	50%
	Navigation was overwhelming	37.5%
	Nothing	37.5%
How would you use this app at your house?	I wouldn't use it at home	62.5%
	Finding things like keys	37.5%
How would you use this app in an unfamiliar place?	At airports	62.5%
	At hospitals	37.5%
	At a mall	12.5%
	In outdoors	12.5%

Table 3: Relevant questions and answers from interviews to participants.

useful in gaining spatial awareness when walking outdoors (12.5%). In an anecdotal instance, a 12-year-old who tested our app, though not part of the study, expressed a desire to prompt the app "Tell me what's across the street."

Discussion

BLV people have developed sophisticated navigation and interaction skills over the years. For example, participant P4 used echolocation via tongue clicks for environmental awareness, while participants P2 and P6 navigated at a slow pace without tools, demonstrating remarkable orientation and mobility abilities. Despite these skills, our system proved particularly useful in challenging situations: when objects were distant or located in unexpected places (e.g., the microwave was at waist height in the opposite corner rather than at the expected head height), and when destinations involved safety considerations (e.g., in task T2, participants did not know whether stairs ascended or descended, or whether a handrail was present). In such cases, our system provided a new way to perceive and interact with the environment.

Overall, the assessment tasks successfully balanced simplicity and complexity: straightforward enough to be achievable without technology, yet challenging enough to benefit from assistance. While our prototype enables users to discover their environment in entirely new ways, this paper focuses specifically on task performance rather than environmental discovery.

We did not include a baseline condition because it would impose an unreasonable burden on participants. In a typical experimental task, a participant begins in a building lobby with instructions to "Go to the conference room and sit in a chair" but the conference room's location was intentionally unknown to test various technological solutions. Without as-

sistive technology, a blind person would need to initiate a time-intensive and frustrating discovery procedure: walking besides walls, testing doors, with the hope of finding the correct room.

Conversational Processing The incorporation of disparate technologies, such as AR spatial anchors and object detection, undoubtedly contributes to an improvement in task performance. However, beyond this, conversation serves as the "connective tissue" of the user experience that, in combination with a cognitive architecture approach, cohesively integrates the other modules into a more comprehensive assistive agent.

From a speech-processing perspective, nuances exist between conversational interaction and navigational guidance. The timing of speech instruction generation before approaching a turning point depends on user speed, waypoint proximity, and instruction duration. Neglecting these aspects may lead to lagged or overlapped instructions, confusing the user, as reported in the error analysis.

Object Interaction During the error analysis, we observed that object interaction was affected by occlusion, proximity, and angle with respect to the object, as well as overfitting of the model, resulting in mislabeling errors. We note that users easily adjusted to these errors. While many of these issues can be addressed by training more powerful computer vision models, the limited resources on the phone introduce a trade-off between running slower but more accurate models vs. faster but less accurate ones. Fortunately, as phone hardware advances, this issue may diminish.

Object Detection vs. Spatial Anchors Object detection and spatial anchors both contribute to user spatial awareness, each with its advantages and drawbacks. Spatial anchors excel in modeling static elements like offices and furniture, while object detection is more adept at capturing dynamic aspects like people and rearranged chairs. In terms of metadata, spatial anchors may store hierarchically organized virtual content, pre-populated for navigation and interaction, whereas object detection provides real-time labels with a semi-structured data format.

User Experience Object detection errors can impact the user experience and task performance, resulting in prolonged completion times and increased clarifying interactions between the user and system. Recurring object detection errors may cause the user to adopt partial disbelief about the system's accuracy.

In a positive user experience, a proof of concept (not part of the measurement scenarios) involved the system injecting contextual information during interaction. For example, while the user navigated to their destination, the system provided relevant surroundings descriptions and recommendations, such as *"In case you're thirsty, there's a drinking fountain six steps away on your left"*. This proactive approach enhanced the user's situational and spatial awareness, enabling them to make opportunistic decisions.

Related Work

To our knowledge, no existing commercial product or scientific work integrates and measures task performance for the three technologies discussed in this paper (real-time object detection, AR spatial anchors, and conversational interaction). A recent work (Kuribayashi et al. 2025), based on a suitcase device, reports on a user study of discovery (called exploration in the work). The study confirms that this combination of produces a desirable experience for user. The study also confirms the issue of machine learning errors. Our work complicates this study by focusing on a cognitive architecture and by reporting task interaction times.

Commercial Products

Commercial products like Orcam MyEye PRO 2.0 (OrCam 2023), Aira (Aira 2023), Be-My-Eyes (Eyes 2023), and Seeing AI (Microsoft 2023a; Granquist et al. 2021) share similarities with our approach in interacting with surrounding objects via object detection. However, distinctions arise as these products have a limited field of view, resulting in challenges in accurately determining the orientation and distances of objects that are not in front of the user. In contrast, our approach computes the spatial position and orientation of anchors around the user, thereby enhancing their awareness of surrounding objects. Unlike our approach, Aira enables video calls for real-time descriptions (Lee et al. 2022). In terms of navigation, Microsoft's Soundscape (Microsoft 2023b), Clew (Lan 2024), and Envision (Envision 2023) convert visual information to speech like our system, but they lack support for back-and-forth, multi-turn conversations and object detection. Clew aligns with our work through AR indoor navigation.

Conversational Processing

From a research perspective, our work draws inspiration from (Vystrcil et al. 2014), which sheds light on the challenges faced by BLV people when a navigational system provides incomplete or incorrect spatial information. In (Chen and Shiu 2020), an assistive outdoor navigational system is introduced, integrating image captioning for scene description and basic conversational interaction, albeit limited to only determining the user's destination. While influenced by NavCog3's interactive pseudo-dialogue experience in indoor navigation (Sato et al. 2017, 2019a), our approach extends further by incorporating interaction with objects and regarding conversation as a fundamental and continuous aspect of the entire navigational experience. MagNav(Giudice et al. 2019) suggests an assistive navigation interface that uses a building's magnetic signatures for user location and guidance. Our system divides conversation into navigation and navigation modes and may benefit from a more complex blending of descriptions provided by recent works such as WorldScribe (Chang, Liu, and Guo 2024) and ChitChatGuide (Kaniwa et al. 2024). These techniques are readily integrated into our cognitive architecture, but our blend of functionality is not easily integrated into the architectures described.

Autonomous robots, such as NavCue (Chen et al. 2016) and CaBot (Guerreiro et al. 2019), have been used for assistive navigation, emphasizing multi-sensory information and object recognition. NavCue utilizes speech guidance and physical gestures for contextual information, while CaBot, a suitcase-shaped robot, uses object recognition for obstacle avoidance with a wider field of view (270°). Several works (Kuribayashi et al. 2025, 2023, 2022; Kubota et al. 2024; Campos et al. 2021) use SLAM and related techniques to build a map in real-time during navigation and provide descriptions of the world for discovery. Mapping during discovery significantly lowers the cost of deployment in a new environment.

Object Interaction

While lacking a conversational aspect, various approaches aid BLV people in object interaction during navigation. Navig (Katz et al. 2012) employs geo-located object models detected by real-time embedded vision algorithms, and Foresee (Zhao et al. 2019c) utilizes techniques like magnification or edge enhancement through smartphones. Access Lens (Kane, Frey, and Wobbrock 2013) can detect text in an image and direct a blind person's finger towards a text target. Some methods utilize depth cameras to perceive the environment, identify structures, and use sonification for obstacle information (Brock and Kristensson 2013; Kanwal et al. 2015; Shangguan et al. 2014). Other research focuses on training deep learning models for improved signage and door recognition during indoor navigation (Afif et al. 2020; Bashiri et al. 2018).

During object interaction, the visual search problem is addressed by enhancing object detection models and involving on-demand human workers to answer visual questions. Prior work (Bigham et al. 2010; Brady et al. 2013; Guo, Chen, and Bigham 2015; Guo et al. 2016) combines on-demand crowdsourcing for labeling pictures taken by BLV people, computer vision techniques for tracking the user's finger pointing at appliance controls, and speech synthesis for reading out crowd-provided labels. Despite mitigating object detection mislabeling, this approach still grapples with challenges tied to the quality of photos taken by users.

Augmented Reality

AR is widely employed in assistive indoor navigation and interaction. CARA (Liu, Stiles, and Meister 2018) captures video data, extracts essential scene knowledge, and conveys it to the user efficiently, recognizing spoken commands and providing descriptions of virtual objects, similar to our approach's use of spatial anchors. Another platform using Hololens Smartglass combines visual and audio wayfinding, offering verbal navigation commands, environmental feedback, visual navigation directions, and obstacle markers (Zhao et al. 2020). While prior work has used AR markers for obstacle localization and positioning, our approach relies on spatial anchor technology with robust computer vision algorithms, eliminating the need for square-based fiducials and ensuring accurate anchor position determination even in full occlusion. Several works focus on visual, audio, or haptic augmentations via AR (Zhao et al. 2019b,a; Guan, Xiong, and Fan 2024; Schieber et al. 2024). However, unlike our approach, they solely use AR either to enhance interaction with objects or to support navigation, not both.

High-precision Localization

Most navigation systems lack granular localization information (Saha et al. 2019). Recent work (van der Bie et al. 2016; Sato et al. 2017; Bai et al. 2014; Ganz et al. 2014; Pérez et al. 2017) focuses on providing information on semantic features of the environment. Such systems demonstrate the usefulness of the landmark-based navigation approach. However, contrasting with the straightforward process of placing and maintaining AR spatial anchors, these approaches require additional instrumentation and maintenance effort to augment the physical environment (e.g., when using RFID sensors (Ganz et al. 2011), NFC tags (Ganz et al. 2014), or Bluetooth beacons (Sato et al. 2017)), and significant cost for setting up databases of floor maps (Fallah et al. 2012; Bai et al. 2014; Fallah et al. 2012; Ganz et al. 2014).

Conclusions

We present an architectural approach and functional prototype that brings together two areas of accessibility research: independent indoor navigation and interaction with objects for people who are blind and low-vision. Our approach combines three technologies: AR spatial anchors, real-time object detection, and LLM-based conversation. We ran an assessment study to evaluate different settings where some features of our system were enabled/disabled. We demonstrate that, overall, the combination and orchestration of the three technologies via a cognitive approach fosters effective navigation and interaction with surrounding objects, improving spatial awareness and task completion times in blind and low-vision users.

Ethics Statement

To address system reliability when cloud-based LLM services become unavailable due to network disruptions or service outages, our architecture incorporates on-device natural language processing fallbacks for critical functions such as intent recognition and entity extraction. A particularly critical safety scenario involves stair navigation during a cloud outage: if the LLM becomes unavailable while a user is approaching stairs, the system pauses navigation instructions, attempts to relocalize the user using available spatial anchors and on-device object detection, and requires explicit confirmation from local perception modules before resuming any directional guidance. This conservative fallback strategy attempts to insure that users are not given potentially unsafe instructions based on incomplete information.

Acknowledgments

The contents of this paper were developed under grants from the National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR grant numbers 90DPGE0003 and 90REGE0007). The authors thank Claudia Folska and Zeeishan Riaz for extensive discussions about accessibility.

References

Afif, M.; Said, Y.; Pissaloux, E.; Atri, M.; et al. 2020. Recognizing signs and doors for Indoor Wayfinding for Blind and Visually Impaired Persons. In *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 1–4. NY, USA: IEEE.

Aira. 2023. We're Aira, a Visual Interpreting Service.

ARKit. 2025. Apple ARKit.

Bai, Y.; Jia, W.; Zhang, H.; Mao, Z.-H.; and Sun, M. 2014. Landmark-based indoor positioning for visually impaired individuals. In *2014 12th International Conference on Signal Processing (ICSP)*, 668–671. NY, USA: IEEE.

Bashiri, F. S.; LaRose, E.; Badger, J. C.; D'Souza, R. M.; Yu, Z.; and Peissig, P. 2018. Object detection to assist visually impaired people: A deep neural network adventure. In *International symposium on visual computing*, 500–510. New York, USA: Springer.

Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B.; White, S.; et al. 2010. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 333–342. NY, USA: ACM.

Brady, E.; Morris, M. R.; Zhong, Y.; White, S.; and Bigham, J. P. 2013. Visual challenges in the everyday lives of blind people. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 2117–2126. NY, USA: SIGCAI.

Branham, S. M.; and Mukkath Roy, A. R. 2019. Reading Between the Guidelines: How Commercial Voice Assistant Guidelines Hinder Accessibility for Blind Users. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19, 446–458. New York, NY, USA: Association for Computing Machinery. ISBN 9781450366762.

Brock, M.; and Kristensson, P. O. 2013. Supporting blind navigation using depth sensing and sonification. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, 255–258. NY, USA: ACM.

Campos, C.; Elvira, R.; Rodríguez, J. J. G.; Montiel, J. M.; and Tardós, J. D. 2021. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6): 1874–1890.

Chang, R.-C.; Liu, Y.; and Guo, A. 2024. WorldScribe: Towards Context-Aware Live Visual Descriptions. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 1–18.

Chen, C.-H.; and Shiu, M.-F. 2020. RNN-based Dialogue Navigation System for Visually Impaired. In *2020 International Conference on Pervasive Artificial Intelligence (ICPAI)*, 140–143. NY, USA: ICPAI.

Chen, H.; Liu, X.; Yin, D.; and Tang, J. 2017. A Survey on Dialogue Systems: Recent Advances and New Frontiers. *SIGKDD Explor. Newsl.*, 19(2): 25–35.

Chen, K.; Plaza-Leiva, V.; Min, B.; Steinfeld, A.; and Dias, M. B. 2016. NavCue: Context Immersive Navigation Assistance for Blind Travelers. In Bartneck, C.; Nagai, Y.; Paiva, A.; and Sabanovic, S., eds., *The Eleventh ACM/IEEE International Conference on Human Robot Interation, HRI 2016, Christchurch, New Zealand, March 7-10, 2016*, 559. NY, USA: IEEE/ACM.

DSTC8. 2021. Google Research DSTC8 Schema Guided Dialogue.

Envision. 2023. Envision glasses.

Eyes, B. M. 2023. Be My AI. AI-powered visual assistance.

Fallah, N.; Apostolopoulos, I.; Bekris, K.; and Folmer, E. 2012. The user as a sensor: navigating users with visual impairments in indoor spaces using tactile landmarks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 425–432. NY, USA: SIGCHI.

Fountain, U. 2020. Results of the 2020 User Testing Industry Report. https://www.userfountain.com/results-of-the-2020-user-testing-industry-report.

Ganz, A.; Gandhi, S. R.; Schafer, J.; Singh, T.; Puleo, E.; Mullett, G.; and Wilson, C. 2011. PERCEPT: Indoor navigation for the blind and visually impaired. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 856–859. NY, USA: IEEE.

Ganz, A.; Schafer, J. M.; Tao, Y.; Wilson, C.; and Robertson, M. 2014. PERCEPT-II: Smartphone based indoor navigation system for the blind. In *2014 36th annual international conference of the IEEE engineering in medicine and biology society*, 3662–3665. NY, USA: IEEE.

Giudice, N. A.; Whalen, W. E.; Riehle, T. H.; Anderson, S. M.; and Doore, S. A. 2019. Evaluation of an Accessible, Real-Time, and Infrastructure-Free Indoor Navigation System by Users Who Are Blind in the Mall of America. *Journal of Visual Impairment & Blindness*, 113(2): 140–155.

Google. 2025. Google MLKit.

GPT2. 2024. Hugging Face API for GPT2.

Granquist, C.; Sun, S. Y.; Montezuma, S. R.; Tran, T. M.; Gage, R.; and Legge, G. E. 2021. Evaluation and Comparison of Artificial Intelligence Vision Aids: Orcam MyEye 1 and Seeing AI. *Journal of Visual Impairment & Blindness*, 115(4): 277–285.

Guan, Z.; Xiong, Z.; and Fan, M. 2024. FetchAid: Making Parcel Lockers More Accessible to Blind and Low Vision People With Deep-learning Enhanced Touchscreen Guidance, Error-Recovery Mechanism, and AR-based Search Support. arXiv:2402.15723.

Guerreiro, J.; Sato, D.; Asakawa, S.; Dong, H.; Kitani, K. M.; and Asakawa, C. 2019. Cabot: Designing and evaluating an autonomous navigation robot for blind people. In *The 21st International ACM SIGACCESS conference on computers and accessibility*, 68–82. NY, USA: ACM.

Guo, A.; Chen, X. A.; and Bigham, J. P. 2015. ApplianceReader: A Wearable, Crowdsourced, Vision-Based System to Make Appliances Accessible. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15,

2043–2048. New York, NY, USA: Association for Computing Machinery. ISBN 9781450331463.

Guo, A.; Chen, X. A.; Qi, H.; White, S.; Ghosh, S.; Asakawa, C.; and Bigham, J. P. 2016. VizLens: A Robust and Interactive Screen Reader for Interfaces in the Real World. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, 651–664. New York, NY, USA: Association for Computing Machinery. ISBN 9781450341899.

Kane, S. K.; Frey, B.; and Wobbrock, J. O. 2013. Access lens: a gesture-based screen reader for real-world documents. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 347–350. New York, NY, USA: Association for Computing Machinery.

Kaniwa, Y.; Kuribayashi, M.; Kayukawa, S.; Sato, D.; Takagi, H.; Asakawa, C.; and Morishima, S. 2024. ChitChatGuide: Conversational Interaction Using Large Language Models for Assisting People with Visual Impairments to Explore a Shopping Mall. *Proceedings of the ACM on Human-Computer Interaction*, 8(MHCI): 1–25.

Kanwal, N.; Bostanci, E.; Currie, K.; and Clark, A. F. 2015. A navigation system for the visually impaired: a fusion of vision and depth sensor. *Applied bionics and biomechanics*, 2015: 221–232.

Katz, B. F.; Kammoun, S.; Parseihian, G.; Gutierrez, O.; Brilhault, A.; Auvray, M.; Truillet, P.; Denis, M.; Thorpe, S.; and Jouffrais, C. 2012. NAVIG: augmented reality guidance system for the visually impaired. *Virtual Reality*, 16(4): 253–269.

Kubota, M.; Kuribayashi, M.; Kayukawa, S.; Takagi, H.; Asakawa, C.; and Morishima, S. 2024. Snap&Nav: Smartphone-based Indoor Navigation System For Blind People via Floor Map Analysis and Intersection Detection. *Proceedings of the ACM on Human-Computer Interaction*, 8(MHCI): 1–22.

Kuribayashi, M.; Ishihara, T.; Sato, D.; Vongkulbhisal, J.; Ram, K.; Kayukawa, S.; Takagi, H.; Morishima, S.; and Asakawa, C. 2023. Pathfinder: Designing a map-less navigation system for blind people in unfamiliar buildings. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–16.

Kuribayashi, M.; Kayukawa, S.; Vongkulbhisal, J.; Asakawa, C.; Sato, D.; Takagi, H.; and Morishima, S. 2022. Corridor-Walker: Mobile indoor walking assistance for blind people to avoid obstacles and recognize intersections. *Proceedings of the ACM on Human-Computer Interaction*, 6(MHCI): 1–22.

Kuribayashi, M.; Uehara, K.; Wang, A.; Morishima, S.; and Asakawa, C. 2025. WanderGuide: Indoor Map-less Robotic Guide for Exploration by Blind People. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–21.

Laird, J. E.; Lebiere, C.; and Rosenbloom, P. S. 2017. A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. *AI Magazine*, 38(4): 13–26.

Lan, B. 2024. Clew: Revolutionary Indoor Navigation for iOS. http://www.clewapp.org/. March 8th, 2024.

Lee, S.; Yu, R.; Xie, J.; Billah, S. M.; and Carroll, J. M. 2022. Opportunities for human-AI collaboration in remote sighted assistance. In *27th International Conference on Intelligent User Interfaces*, 63–78. NY, USA: IEEE.

Liu, Y.; Stiles, N. R.; and Meister, M. 2018. Augmented reality powers a cognitive assistant for the blind. *ELife*, 7: e37841.

LlAVA. 2024. LlAVA Foundational Model API.

Microsoft. 2023a. Seeing AI App from Microsoft.

Microsoft. 2023b. Soundscape from Mircosoft Research.

Mistral. 2024. Mistral Foundational Model API.

Nielsen, J.; and Landauer, T. K. 1993. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, 206–213. New York: ACM Press.

Ollama. 2024. Ollama API.

OrCam. 2023. Orcam Myeye 2.0 - for people who are blind or visually impaired.

Pérez, J. E.; Arrue, M.; Kobayashi, M.; Takagi, H.; and Asakawa, C. 2017. Assessment of semantic taxonomies for blind indoor navigation based on a shopping center use case. In *Proceedings of the 14th International Web for All Conference*, 1–4. NY, USA: ACM.

Romero, O. J.; Wang, A.; Zimmerman, J.; Steinfeld, A.; and Tomasic, A. 2021. A Task-Oriented Dialogue Architecture via Transformer Neural Language Models and Symbolic Injection. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 438–444. Singapore and Online: Association for Computational Linguistics.

Romero, O. J.; Zimmerman, J.; Steinfeld, A.; and Tomasic, A. 2023. Synergistic integration of large language models and cognitive architectures for robust ai: An exploratory analysis. In *Proceedings of the AAAI Symposium Series*, volume 2, 396–405. Washington, D.C.: AAAI Press.

Saha, M.; Fiannaca, A. J.; Kneisel, M.; Cutrell, E.; and Morris, M. R. 2019. 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*, 222–235. NY, USA: ACM.

Sato, D.; Oh, U.; Guerreiro, J.; Ahmetovic, D.; Naito, K.; Takagi, H.; Kitani, K. M.; and Asakawa, C. 2019a. NavCog3 in the wild: Large-scale blind indoor navigation assistant with semantic features. *ACM Transactions on Accessible Computing (TACCESS)*, 12(3): 1–30.

Sato, D.; Oh, U.; Guerreiro, J. a.; Ahmetovic, D.; Naito, K.; Takagi, H.; Kitani, K. M.; and Asakawa, C. 2019b. NavCog3 in the Wild: Large-Scale Blind Indoor Navigation Assistant with Semantic Features. *ACM Trans. Access. Comput.*, 12(3).

Sato, D.; Oh, U.; Naito, K.; Takagi, H.; Kitani, K.; and Asakawa, C. 2017. Navcog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment. In *Proceedings*

of the 19th International ACM SIGACCESS Conference on Computers and Accessibility, 270–279. NY, USA: ACM.

Schieber, H.; Kleinbeck, C.; Theelke, L.; Kraft, M.; Kreimeier, J.; and Roth, D. 2024. MR-Sense: A Mixed Reality Environment Search Assistant for Blind and Visually Impaired People. In *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, 166–175. New York, NY, USA: Association for Computing Machinery.

Shangguan, L.; Yang, Z.; Zhou, Z.; Zheng, X.; Wu, C.; and Liu, Y. 2014. Crossnavi: enabling real-time crossroad navigation for the blind with commodity phones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 787–798. Seattle, USA: ACM.

SpatialAnchors. 2024. Microsoft Spatial Anchors.

TensorFlow. 2025. TensorFlow Toolkit.

van der Bie, J.; Visser, B.; Matsari, J.; Singh, M.; Van Hasselt, T.; Koopman, J.; and Kröse, B. 2016. Guiding the visually impaired through the environment with beacons. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, 385–388. NY, USA: ACM.

Vystrcil, J.; Maly, I.; Balata, J.; and Mikovec, Z. 2014. Navigation Dialog of Blind People: Recovery from Getting Lost. In Dalmas, T.; Götze, J.; Gustafson, J.; Janarthanam, S.; Kleindienst, J.; Mueller, C.; Stent, A.; and Vlachos, A., eds., *Proceedings of the EACL 2014 Workshop on Dialogue in Motion*, 58–62. Gothenburg, Sweden: Association for Computational Linguistics.

Yao, J.-Y.; Ning, K.-P.; Liu, Z.-H.; Ning, M.-N.; Liu, Y.-Y.; and Yuan, L. 2024. LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples. arXiv:2310.01469.

Zhao, Y.; Cutrell, E.; Holz, C.; Morris, M. R.; Ofek, E.; and Wilson, A. D. 2019a. SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, 1–14. New York, NY, USA: Association for Computing Machinery. ISBN 9781450359702.

Zhao, Y.; Kupferstein, E.; Castro, B. V.; Feiner, S.; and Azenkot, S. 2019b. Designing AR Visualizations to Facilitate Stair Navigation for People with Low Vision. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, 387–402. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368162.

Zhao, Y.; Kupferstein, E.; Rojnirun, H.; Findlater, L.; and Azenkot, S. 2020. The Effectiveness of Visual and Audio Wayfinding Guidance on Smartglasses for People with Low Vision. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, 1–14. New York, NY, USA: Association for Computing Machinery. ISBN 9781450367080.

Zhao, Y.; Szpiro, S.; Shi, L.; and Azenkot, S. 2019c. Designing and Evaluating a Customizable Head-Mounted Vision Enhancement System for People with Low Vision. *ACM Trans. Access. Comput.*, 12(4).