

Towards Human-AI Symbiotic Spatial Perception

Xia Su

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:

Jon E. Froehlich, Chair

Ranjay Krishna

Yang Zhang

Program Authorized to Offer Degree:

Computer Science and Engineering

© Copyright 2026

Xia Su

University of Washington

Abstract

Towards Human-AI Symbiotic Spatial Perception

Xia Su

Chair of the Supervisory Committee:

Professor Jon E. Froehlich
Computer Science and Engineering

Artificial intelligence has become an indispensable part of digital life. People routinely use AI to write, program, search, summarize, and make decisions. Yet this emerging human-AI symbiosis remains concentrated in the digital realm, where people work primarily with language through digital devices. It is far less developed in the physical world, where people navigate and act under real stakes. People must still move through unfamiliar places, manipulate objects, assess risks, and make decisions through their own situated perception and at the cost of their own time and effort.

This limitation does not arise from a lack of AI research on the physical world. Over several decades, computer vision, robotics, augmented reality, and embodied AI have produced systems that recognize objects, reconstruct environments, navigate, plan motion, and increasingly predict how real-world scenes may evolve. The challenge is to translate these capabilities into assistance that fits human activity. Spatial evidence is viewpoint-dependent, incomplete, multimodal, and subject to change. Its usefulness also depends on who will act, what that person is trying to accomplish, and how an inferred result is communicated. A model can therefore perform well on a general benchmark while remaining too slow, insufficiently grounded, mismatched to a person's capabilities, or too opaque for a consequential spatial decision.

This dissertation advances a vision of *human-AI symbiotic spatial perception*: a perceptual partnership in which people and AI systems combine complementary knowledge and capabilities to acquire, interpret, and communicate information about physical space. I argue that useful spatial assistance depends on de-

signing AI around both people and the physical world. Such systems must deliver timely, in-context output; acquire task-relevant spatial evidence beyond a person’s direct presence; align spatial judgments with human capabilities and goals; and represent spatial data in forms that people can perceive, inspect, and manipulate. The dissertation investigates these requirements through four research frontiers and six projects.

Frontier I, In-Context, Real-Time Personalized Spatial Assistance, examines RASSAR, a mobile augmented-reality system for accessibility and safety assessment. Frontier II, Spatial Perception Beyond Physical Presence, explores automated and human-guided spatial capture through RAIS and FlyMeThrough. Frontier III, Aligning AI Spatial Reasoning with Human Needs, introduces CapNav, a benchmark of capability-conditioned indoor navigation. Frontier IV, Human-Centered Spatial Representation and Interaction, investigates object- and event-centered techniques for spatial authoring through SonifyAR and DepthScape.

Across the projects, the findings show that useful assistance can emerge even when AI capabilities remain imperfect. RASSAR demonstrates the feasibility of mobile, stakeholder-informed accessibility scanning while exposing the need for personalization and human verification. RAIS and FlyMeThrough show that robots and drones can extend the reach of spatial capture, but that predefined task criteria or sparse human annotations are needed to preserve relevance. CapNav reveals that strong vision-language models degrade when navigation depends on embodiment-specific constraints, metric estimates, and evidence integrated across views. SonifyAR and DepthScape show how intermediate representations, including contextual event descriptions and parametric spatial anchors, let people inspect and shape AI-generated outputs. Together, these contributions establish a human-centered design agenda for spatial AI that is timely, far-reaching, capability-aligned, expressive, and accountable to the people who use it.

Acknowledgements

Beginning this PhD was a bold move. Coming from an architecture background, I could not have anticipated how challenging, surprising, and fulfilling a PhD in computer science would be. With the guidance, support, and love of many people, I found my way through that uncertainty and, at last, to this dissertation.

This work would not have been possible without my advisor, Prof. Jon E. Froehlich, who welcomed me into computer science and HCI and patiently supported my academic growth over the past five years. His intellectual energy, discipline, creativity, and genuine enthusiasm fueled my development as a researcher and sustained me through periods of difficulty and confusion. I am also fortunate to have met other generous mentors along the way: Prof. Liang He, who first introduced me to the Makeability Lab; Prof. Chang Xiao, who guided my first internship at Adobe and became a close friend ever since; and Dr. Cuong Nguyen, whose mentorship and support I deeply benefited from and appreciate.

I am grateful to my project collaborators, from whom I learned an enormous amount: Dr. Han Zhang, Dr. Kaiming Cheng, Wyatt Olson, Dr. Eunye Koh, Ruiqi Chen, Prof. Yang Zhang, William Huang, Jingwei Ma, Benlin Liu, Dr. Matheus Gadelha, and Prof. Ranjay Krishna.

I thank the members of the Makeability Lab, who created an inspiring academic environment and brought me both support and joy: Daniel Campos Zamora, Chu Li, Jaewook Lee, Arnavi Chheda-Kothary, Jared Hwang, and Mikey Saugstad. I am equally grateful to the members of the UbiComp Lab with whom I shared so many memorable moments, including Shirley Xue, Richard Li, Nancy Yujia Liu, Guilin Hu, Maggie Shao, and many others.

I am fortunate to have friends who supported and encouraged me throughout these years: Dr. Xin Chen, Jingwei Ma, Dr. Daogao Liu, Maochuan Zhang, Tuochao Chen, Dr. Sirui Lu, Dr. Linxing Preston Jiang, Dr. Zekun Shao, Bingxin Wang, Dr. Xiaohai Hu, Kaiyuan Liu, Bryce Yicong Chen, Weihang Xu, Zonglin

Di, and Jiwon Chung.

I want to thank my parents, who have been extraordinary role models and a source of unconditional support and happiness, whether I was thriving or struggling.

Finally, I express my deepest gratitude to my partner who, despite the distance and time difference, supported me with great care and brought me immeasurable joy.

DEDICATION

To my parents, who have given me boundless courage, strength, and assurance.

Contents

1	Introduction	1
1.1	Frontier I: In-Context, Real-Time Personalized Spatial Assistance	3
1.2	Frontier II: Spatial Perception Beyond Physical Presence	3
1.3	Frontier III: Aligning AI Spatial Reasoning with Human Needs	4
1.4	Frontier IV: Human-Centered Spatial Representation and Interaction	4
1.5	Summary of Contributions	5
1.6	Chapter Overview	7
2	Overview of Human and AI Spatial Perception	9
2.1	Human Spatial Perception	9
2.1.1	Perception and Action in an Embodied World	10
2.1.2	Organizing Spatial Knowledge	10
2.1.3	Sources, Scales, and Limits of Spatial Knowledge	11
2.2	AI World Understanding	12
2.2.1	From Geometric Mapping to Learned Scene Representations	13
2.2.2	Multimodal and Embodied Spatial Intelligence	13
2.2.3	Predictive World Models	14
2.2.4	Limits in AI Spatial Grounding	15
2.3	Human-AI Interaction Frameworks	16
2.4	Combining AI World Understanding with Human Needs	18
2.4.1	Accessible Navigation and Nonvisual Spatial Cognition	18

2.4.2	Accessibility Mapping and Environmental Auditing	19
2.4.3	Situated and Remote Spatial Assistance	20
3	Frontier I: In-Context, Real-Time Personalized Spatial Assistance	23
3.1	Preamble	23
3.2	RASSAR: Room Accessibility and Safety Scanning in Augmented Reality	24
3.2.1	Introduction	24
3.2.2	Design Process	27
3.2.3	The RASSAR System	31
3.2.4	Study 2: Technical Evaluation	38
3.2.5	Study 3: User Study	42
3.2.6	Discussion	45
3.2.7	Conclusion	48
3.3	Frontier Synthesis	48
4	Frontier II: Spatial Perception Beyond Physical Presence	49
4.1	Preamble	49
4.2	RAIS: Robotic Accessibility Indoor Scanning	50
4.2.1	Introduction	50
4.2.2	The RAIS Prototype	52
4.2.3	Preliminary Evaluation	54
4.2.4	Discussion and Future Work	55
4.2.5	Conclusion	56
4.3	From Room-Scale Auditing to Large-Space Semantic Mapping	56
4.4	FlyMeThrough: Human-AI Collaborative Indoor Mapping	57
4.4.1	Introduction	57
4.4.2	The 3D Mapping of Indoor Spaces	59
4.4.3	The FlyMeThrough Interface	65
4.4.4	Evaluation	68

4.4.5	Discussion and Future Work	75
4.4.6	Conclusion	77
4.5	Frontier Synthesis	77
5	Frontier III: Aligning AI Spatial Reasoning with Human Needs	79
5.1	Preamble	79
5.2	CapNav: Capability-Conditioned Indoor Navigation	80
5.2.1	Introduction	80
5.2.2	The CapNav Benchmark	83
5.2.3	Experiments	89
5.2.4	Findings	93
5.2.5	Discussion	95
5.2.6	Conclusion	97
5.3	Frontier Synthesis	97
6	Frontier IV: Human-Centered Spatial Representation and Interaction	99
6.1	Preamble	99
6.2	SonifyAR: Context-Aware Sound Generation in Augmented Reality	100
6.2.1	Introduction	101
6.2.2	The Design of SonifyAR	103
6.2.3	SonifyAR Implementation	106
6.2.4	User Study	112
6.2.5	Applications	113
6.2.6	Discussion and Future Work	119
6.2.7	Conclusion	121
6.3	From Context-Aware Output to Manipulable Spatial Structure	121
6.4	DepthScape: Parametric Anchors for 2.5D Visual Design	122
6.4.1	Introduction	122
6.4.2	Designing DepthScape	125

6.4.3	The DepthScape System	129
6.4.4	Evaluation	136
6.4.5	Application Scenarios	142
6.4.6	Discussion and Future Work	143
6.4.7	Conclusion	147
6.5	Frontier Synthesis	147
7	Synthesis: Toward Human-AI Symbiotic Spatial Perception	149
7.1	Contributions	149
7.1.1	A Four-Frontier Framework	149
7.1.2	System, Interaction, and Benchmark Contributions	150
7.1.3	Evidence About Designing with Imperfect AI	151
7.2	Cross-Frontier Lessons	151
7.3	Users, Stakeholders, and Scope of Evidence	153
7.4	Current Deployment Constraints	155
7.5	Future Directions	158
8	Conclusion	161

List of Figures

1.1	The ecosystem of human-AI symbiotic spatial perception developed in this dissertation. Across four frontiers and six projects, the framework extends beyond a person, an AI model, and the physical environment to include sensing devices that capture evidence beyond direct physical reach, field experts who contribute local knowledge, interaction techniques that support effective communication, and alignment mechanisms that connect AI outputs to human capabilities and goals.	5
3.1	RASSAR is a mobile AR application for semi-automatically identifying, localizing, and visualizing indoor accessibility and safety issues. It scans a home using LiDAR and computer vision, evaluates detected elements against accessibility and safety rubrics, and presents an interactive 3D summary.	25
3.2	Design probes for RASSAR’s interaction and interface design. For each question, we provide three or four options as interface mock-ups.	32
3.3	RASSAR system overview. RASSAR (1) scans and reconstructs a home space, (2) detects accessibility and safety issues in the space, and (3) visualizes the results in real time through AR.	32

3.4	Informed by literature and our formative study, RASSAR can detect 20 types of accessibility and safety issues across four categories: object dimension, object position, risky item, and lack of assistive item. Black icons indicate the accessibility communities for which each issue is relevant. Safety and accessibility needs are fluid and shaped not only by disability but also by fatigue and time of day, so individuals may not map exactly to these categories. The custom JSON-based rubric could support more precise individual specification in the future, for example through a custom authoring interface.	35
3.5	RASSAR’s AR-based scanning interface and a detailed view of a detected issue. (1) During a scan, RASSAR shows detected problems in real time through AR overlays, including (a) red spheres that users can select to view more information and confirm or delete a detection, and (b) computer-vision detections with green bounding boxes, text labels, and confidence scores. A gray bounding box (c) indicates the computer-vision field of view. A minimap (d) adapts to the user’s orientation and position and shows reconstruction results in real time. (2) The user can select an identified issue (a) to view more information and recommended solutions and can confirm or delete the issue (e).	37
3.6	Three examples of RASSAR’s post-scan interactive summary. A user can interact with the 3D reconstruction, inspect detailed information about objects or issues, and remove issues that are erroneous or do not apply. The button in the upper-right corner exports scan results to a JSON file.	38
3.7	RASSAR’s technical evaluation performance over ten indoor spaces. (Top) A scatter plot of precision and recall over 30 scans in 10 home spaces. (Bottom) A histogram of Krippendorff’s alpha values across 10 home spaces (3 scans each).	41
3.8	Study 3 participants conducting RASSAR scans.	43
4.1	RAIS combines commodity robotic hardware with mobile sensing to automate indoor mapping and accessibility assessment. The system produces a navigation map, captures a parametric room model, and marks detected accessibility issues.	51

4.2	Example of RAIS’s A* routing. Grid cells measure 20 cm × 20 cm. Black represents walls, gray represents furniture, blue represents scanned space, orange represents the path, and purple represents stopping positions.	53
4.3	Preliminary RAIS evaluation in three indoor spaces: a makerspace (S1), an office (S2), and a classroom (S3). For each space, the figure shows a photograph, the automatically generated 2D map, and RAIS’s scanning route.	55
4.4	FlyMeThrough reconstructs large indoor spaces from commodity-drone video. Sparse user annotations are propagated across frames, localized in the 3D model, and exposed as interactive facility anchors.	57
4.5	Drone-based indoor scanning and reconstruction. (A) The DJI Avata drone. (B) We collect flight footage as 4K, 30-FPS video. (C) We extract frames at 2 FPS. (D) Agisoft Metashape estimates the camera extrinsics of extracted frames and (E) reconstructs the space in 3D. . .	60
4.6	SAM2 segments key indoor facilities. When a user annotates one frame, the mask is propagated to subsequent frames.	61
4.7	Depth-guided ray casting localizes user annotations. (A) Each annotation mask is combined with (B) an inferred depth map to produce (C) a 3D point cloud, which is iteratively enlarged until it intersects the reconstructed model.	63
4.8	The annotation interface. (A) The results panel lists confirmed annotations. (B) The canvas displays selected frames and supports click-based segmentation. (C) The annotation panel provides object types and commands such as “Clear Points,” “Confirm Annotation,” and “Finish Annotation.” (D) The frames panel lists available video frames. (E) After the user confirms an annotation, the server propagates its segmentation mask to subsequent frames, where results appear as bounding boxes. (F) The interface summarizes the completed annotations.	66
4.9	The review interface. (A) A list of available spaces. (B) A 3D viewport displaying the map and annotated objects. (C) A list of annotated facilities that users can highlight or select with the “View” button. Additional controls return the camera to a bird’s-eye view or display an orthographic floor-plan view.	67

4.10	Nine of the scanned spaces.	68
4.11	Three reconstruction failure modes. (A) Two images from S9, a multilevel office atrium whose reconstruction failed. (B) Blurring in a sparsely captured region of a reconstructed map. (C) Missing reconstructed geometry for a waste container visible in the source footage.	71
5.1	CapNav evaluates capability-conditioned navigation from a tour video, navigation-graph nodes, an agent profile, and a navigation task. Outputs are assessed for feasibility, path validity, route traversability, and reasoning validity.	81
5.2	The CapNav benchmark evaluates whether VLMs account for differences in agent mobility when generating navigation plans. In this example, route feasibility and the appropriate path differ across agents.	81
5.3	Overview of CapNav data construction. Starting from a 3D indoor scan, we manually record a touring video and construct a navigation graph. We then use Gemini to generate natural-language navigation tasks. Finally, we annotate traversability for each task and agent by manually controlling virtual agents in the annotation interface.	84
5.4	Examples of CapNav agent profiles. Left: each profile specifies physical dimensions and functional capabilities. Right: virtual agents with the specified dimensions are rendered and maneuvered in the 3D scenes to help verify traversability.	86
5.5	Model accuracy by obstacle type.	92
5.6	CapNav performance across visual-frame-budget settings.	94
6.1	SonifyAR observes an AR event and its user, virtual-object, and physical-scene context, then recommends, retrieves, generates, or transforms a matching sound that can be tested in situ.	101
6.2	Sound-producing opportunities in the triad of User, Virtuality, and Reality.	104
6.3	Overview of the SonifyAR pipeline. The system monitors and logs AR-event context, including the event type, participating virtual and real-world entities, and their attributes such as material. It compiles this information into a text template, which an LLM controller processes to acquire sound assets. The resulting options are presented in a selection panel.	106

6.4	SonifyAR’s event textualization. Left: a user taps a virtual cup through the SonifyAR interface. Right: SonifyAR’s PbD framework extracts a textual representation of the event context.	107
6.5	SonifyAR’s sound acquisition pipeline.	109
6.6	SonifyAR’s authoring interface. Left: SonifyAR’s phone-based AR interface. Right: SonifyAR’s authoring panel.	111
6.7	Rating questions and response distributions from the SonifyAR user study.	114
6.8	An AR physics experiment sonified by SonifyAR. (A) The ramp, metal-ball models, and simulation sequence. (B, C) Contextual information for two collisions in the simulation. (D) Sound assets acquired for the collisions.	115
6.9	A virtual robot walking on different surfaces with different sounds. (A, B) The robot on a wooden surface and the corresponding context. (C, D) The robot on carpet and the corresponding context. (E, F) The acquired sound assets.	116
6.10	An AR dinosaur sonified by SonifyAR. (A) The AR Camera application shows a virtual dinosaur playing an eating animation. (B) Contextual information describes the scene. (C) SonifyAR produces a text prompt and sound output.	117
6.11	Sonification of window placement on Apple Vision Pro. (A) A user places a virtual window on a sofa. (B) Contextual information describes the action. (C) SonifyAR produces a text prompt and sound output.	118
6.12	WoZ example of SonifyAR in the Reality Composer authoring process. (A) A machine model in the authoring interface. (B) Reality Composer’s interface for AR sound. (C) Contextual information for the event in B. (D, E) Sound assets generated and retrieved through SonifyAR. (F) The acquired sounds appear as recommended options.	119
6.13	DepthScape reconstructs latent depth from a single image, extracts AI-recommended parametric anchors, and lets users place and manipulate design elements in the resulting 2.5D space.	123
6.14	Gallery of designs created with DepthScape. Icons in the top row show how each design is formed from one or more planar, cylindrical, or spherical parametric anchors.	123

6.15	Proof-of-concept prototype interface.	126
6.16	In the formative user study, participants replicated two designs (left) using the provided assets (middle) and the proof-of-concept prototype. Example results are shown on the right. .	127
6.17	Design results of the open-ended exploration of the formative user study.	127
6.18	DepthScape’s 3D-space composition and rendering results. Left: a 3D space viewed from a different angle, showing the original input image, reconstructed point cloud, and added design elements. The gray dashed line indicates the camera position. Right: rendered results with design elements embedded in the input-image scene.	130
6.19	DepthScape currently supports three types of parametric anchors that can be directly manipulated through mouse movement. From left to right: planar, cylindrical, and spherical. . . .	132
6.20	DepthScape employs a vision-language model to (A) generate visual programs from (B) input images. The program guides a cylindrical-anchor extraction process, including (C) masking the prompt <code>the train in the image</code> and (D) selecting the corresponding point cloud. The final result is (E) an interactive cylinder precisely aligned with the train. . .	134
6.21	DepthScape’s user interface. (A) The upload control and input image. (B) AI-suggested parametric anchors. (C) Layers added to the scene, with controls to select, hide, show, or remove them. (D) The direct-manipulation canvas. (E) The parameter panel during layer initialization. (F) Additional controls available after initialization, including content rotation and the addition of (G) image content. (H) Fine-tuning controls for layer parameters. . . .	135
6.22	Four types of test images.	136
6.23	Example design results shown to experts.	137
6.24	Four types of parametric anchors. Each example shows the name of the parametric anchor on top.	139
6.25	The distribution of processing time.	139
6.26	Failure cases. (A) Three images for which DepthScape generated no parametric anchor. (B) Reconstruction of a transparent glass surface loses the geometry behind the glass. (C) A request to mask “the cake slice” selects the entire cake. (D) Cylinder fitting selects a horizontal axis instead of the shorter, semantically appropriate vertical axis.	140

6.27 DepthScape adds content, including a rug, wall art, and advertising graphics, to photographs of physical scenes. 143

6.28 Simulated AR effects, including the plane and text for the tea set, and also the virtual UI located behind the sliding doors, in a real-world indoor scene. 144

6.29 DepthScape applies to sketched scenes, supporting rapid storyboarding with visual assets of varying fidelity. 145

List of Tables

3.1	Demographics of the 18 formative-study participants. PID denotes participant ID; “Older adults” identifies participants aged 65 or older; “Children” identifies families with children from birth through age three; BLV denotes participants who are blind or have low vision; OT denotes occupational therapists; and CG denotes caregivers. Roles and identities are not mutually exclusive.	29
3.2	Numbers of images and annotations used to train RASSAR’s custom YOLO model. Because some images contained multiple objects, the annotation count exceeds the image count. . . .	34
3.3	Characteristics of the ten home spaces and RASSAR’s performance in each. Scanning time is reported in seconds, and each performance metric is averaged across three scans of the space.	40
3.4	RASSAR detection performance by accessibility and safety issue. GT denotes the ground-truth count from manual audits; TP, FP, and FN denote true positives, false positives, and false negatives, respectively.	41
3.5	Sources of errors in the RASSAR technical evaluation.	42
3.6	Participant demographics, scan performance, and ratings in the RASSAR user study. Ratings use a seven-point scale; N/A indicates that the participant did not provide a rating. . . .	44
4.1	RAIS test spaces and results. Accessibility issues are identified by their index in Section 4.2.2.3.	55
4.2	Overview of spaces, video data, and reconstruction status.	69
4.3	Participant demographics and annotation performance.	70

5.1 Ratios of feasible tasks and traversable edges for each agent type. 87

5.2 CapNav performance across 13 VLMs. Each value is the best performance across all frame settings tested for that model. The left block reports task metrics (Feas-F1, PV, RTA, RV, and CapNav). The right block reports composite scores for five agent types: adults without motor disabilities, humanoid robots, quadrupedal robots, sweeping robots, and wheelchair users. Bold indicates the best value in each block. Blue indicates open-source models, green indicates proprietary models, and orange indicates models designed for spatial reasoning. N/A indicates that the model did not produce a valid value for the metric. 90

6.1 Expert-review participant demographics. 138

Chapter 1

Introduction

AI has become essential infrastructure for our digital lives. People use AI systems to draft documents, write and debug code, retrieve information, summarize complex material, and support decisions. These interactions increasingly resemble a form of human-AI symbiosis in the digital realm: people contribute goals, judgment, and context, while AI contributes computational scale, pattern recognition, and generative capability.

This human-AI symbiosis, anticipated more than half a century ago in Licklider’s vision of “man-computer symbiosis” [Licklider, 1960], is far less developed in the physical world. Everyday spatial decisions are situated, embodied, and consequential. People must determine where to go, which paths are feasible, what objects can be used, and what risks an action entails. They make these judgments from partial observations, remembered experiences, maps and descriptions, and active exploration, often at the cost of their own time, effort, and exposure.

This limitation does not arise from a lack of AI research on the physical world itself. Large-scale visual datasets and deep neural networks established broad object recognition and segmentation across images and videos [Deng et al., 2009; Krizhevsky et al., 2012; Ravi et al., 2024a]. Simultaneous localization and mapping systems reconstruct geometry while estimating a camera or robot’s position [Mur-Artal and Tardós, 2017], and route planners connect spatial representations to feasible paths [Hart et al., 1968; Savva et al., 2019]. Embodied multimodal and vision-language-action models further connect language and visual observations to plans and robot actions [Driess et al., 2023; Zitkovich et al., 2023]. More recently, predictive

models have begun to simulate action-conditioned future observations for navigation and control [Bar et al., 2025].

These advances are substantial, but benchmarked capabilities do not automatically become useful assistance in everyday physical activity. Spatial evidence is incomplete and viewpoint-dependent; environments and objects change; an answer may arrive too late for the action it is meant to support; and a judgment that is valid for one body, machine, or task may be unsafe for another. The system must also express its interpretation through a modality and representation that a person can understand, question, and correct. Improving a model in isolation addresses only part of this translation problem.

This dissertation studies the human-centered gap between AI world understanding and human spatial perception. I ask how AI can extend what people are able to perceive about the physical world while remaining aligned with a person’s current context, bodily capabilities, goals, and preferred forms of interaction. I call this goal *human-AI symbiotic spatial perception*: a perceptual partnership in which people and AI systems combine complementary knowledge and capabilities to acquire, interpret, and communicate information about physical space. The dissertation advances the following thesis:

Human-AI symbiotic spatial perception depends on designing AI systems around both people and the physical world: systems must deliver in-context output in real time relative to the supported task, acquire task-relevant spatial data beyond a person’s direct presence, align spatial reasoning with human capabilities and goals, and represent spatial data in forms people can perceive and manipulate. Across six projects, this dissertation shows that these design principles can enable useful real-world spatial assistance even when the underlying AI capabilities remain imperfect.

Throughout this dissertation, *real time* is task-relative rather than a universal latency threshold. A spatial system operates in real time when its end-to-end feedback, including sensing, inference, transmission, and presentation, arrives within the *decision horizon* of the ongoing action it supports. Continuous locomotion or collision avoidance may require subsecond responses; interactive scanning may tolerate longer computation when incremental guidance arrives before the user chooses the next viewpoint; and deliberative planning may remain useful over seconds or minutes without being real-time control. I use *timely* more broadly for information that arrives when it can still support a task, including tasks that are not continuously unfolding.

Section 7.4 returns to these distinctions with measured latencies from the dissertation projects.

This research goal extends beyond the scope of any single dissertation. Rather than claiming to solve it, this dissertation develops and evaluates approaches toward this vision through four research frontiers. They are not independent stages in a fixed pipeline. Each identifies a condition that can determine whether AI-generated spatial information becomes useful to a person, and the projects can address more than one condition at once.

1.1 Frontier I: In-Context, Real-Time Personalized Spatial Assistance

The first frontier concerns *immediacy*: can AI provide useful spatial assistance while a person is acting in the environment, with feedback grounded in the present scene and personalized to the user’s needs? Such a system must acquire local evidence, interpret it in real time, relate it to the user’s task and capabilities, and communicate results without demanding excessive attention or expertise.

RASSAR [Su et al., 2024d, 2023, 2022] serves as the example project for this frontier. RASSAR uses a mobile phone’s camera and LiDAR sensor, real-time computer vision, customizable accessibility rubrics, and augmented reality to identify and communicate potential accessibility and safety issues. The project demonstrates both the promise and difficulty of situated assistance: the system must guide scanning, recognize objects at different scales, connect measurements to individual concerns, and expose detections with efficient visualization.

1.2 Frontier II: Spatial Perception Beyond Physical Presence

The second frontier concerns *reach*: can AI help people acquire spatial information beyond their physical presence in space and time? Human perception is constrained by travel, access, risk, memory, and the timing of an observation. Cameras, robots, and drones can extend this reach, but captured geometry or video is not automatically useful. A remote representation must preserve the objects, relations, and functional details required by its intended users.

RAIS [Su et al., 2024a] and FlyMeThrough [Su et al., 2025a] examine complementary approaches. RAIS combines RASSAR with commodity robotic hardware to explore automated indoor accessibility

scanning. It asks whether a robot can independently and regularly establish room models and accessibility assessments. FlyMeThrough addresses larger and more varied indoor spaces through commodity-drone video, 3D reconstruction, and human-AI collaborative annotation. Building managers and occupants identify meaningful points of interest, while AI propagates their annotations and localizes them in a 3D map. Together, the projects show that extending perception beyond physical presence requires both autonomous or semi-autonomous capture and human guidance about what information matters.

1.3 Frontier III: Aligning AI Spatial Reasoning with Human Needs

The third frontier concerns *alignment*: does an AI system interpret the environment in a way that is valid for the particular person or agent relying on it? Spatial perception is subject-conditioned. Stairs, thresholds, door widths, turning spaces, and the ability to operate a door affect bodies differently. AI model outputs, which are based on general training data, can therefore be geometrically plausible and linguistically convincing while still being unusable or unsafe for a certain embodiment.

CapNav [Su et al., 2026a] serves as the example project for this frontier. CapNav benchmarks vision-language models on indoor navigation tasks conditioned on five human and robotic capability profiles. It evaluates not only whether a model predicts feasibility, but whether it constructs a valid path, respects edge-level traversability, and gives an appropriate reason when a route is infeasible. CapNav reveals systematic weaknesses in capability-conditioned reasoning, particularly when models must estimate dimensions or integrate evidence across many views. The project turns spatial alignment from a broad aspiration into a measurable technical and human-centered problem.

1.4 Frontier IV: Human-Centered Spatial Representation and Interaction

The fourth frontier concerns *expressiveness*: how should humans and AI exchange spatial knowledge? AI systems commonly encode space as meshes, masks, point clouds, embeddings, graphs, or generated text. Humans may instead think through objects, materials, landmarks, sounds, depth cues, and direct manipulation. Effective spatial input and output require representations that bridge machine inference and human intuition.

SonifyAR [Su et al., 2024c] and DepthScape [Su et al., 2026b] explore this frontier through different modalities. SonifyAR converts interactions among users, virtual objects, and physical surfaces into textual event representations, then retrieves or generates context-aware sounds. It demonstrates how sound can communicate material and event relationships that are difficult to author through conventional trigger interfaces. DepthScape reconstructs latent depth from a single image and exposes AI-inferred geometry through parametric anchors that designers can directly manipulate. It demonstrates how object-centered interaction can make inherent spatial information inspectable and useful without requiring the user to become a 3D modeling expert. Together, these projects position representation and interaction as core components of spatial intelligence.

1.5 Summary of Contributions

This dissertation makes four overarching contributions:

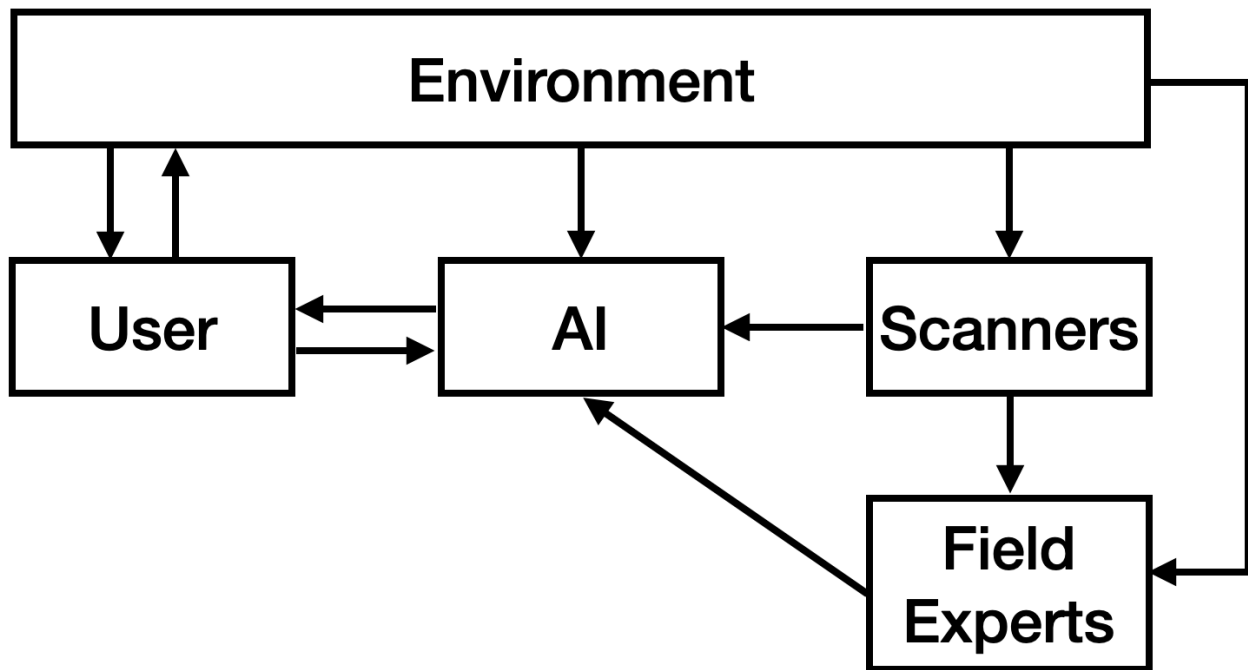


Figure 1.1: The ecosystem of human-AI symbiotic spatial perception developed in this dissertation. Across four frontiers and six projects, the framework extends beyond a person, an AI model, and the physical environment to include sensing devices that capture evidence beyond direct physical reach, field experts who contribute local knowledge, interaction techniques that support effective communication, and alignment mechanisms that connect AI outputs to human capabilities and goals.

1. **A human-centered framework for spatial AI.** It defines human-AI symbiotic spatial perception and develops four design requirements for translating AI world understanding into assistance: timely and personalized support in context, acquisition of evidence beyond direct physical presence, alignment with the capabilities and goals of the person who will act, and human-centered representation and interaction. This framework shifts the unit of analysis from model output alone to the complete ecosystem among people, physical environments, AI inference, interfaces, external scanning devices, and field experts. See Figure 1.1.
2. **Systems and interaction techniques for acquiring and using spatial data.** Across RASSAR, RAIS, FlyMeThrough, SonifyAR, and DepthScape, the dissertation contributes mobile and robotic accessibility scanning, human-guided drone mapping, click-based segmentation propagation and depth-guided raycasting, context-aware sound authoring through programming by demonstration, depth-based blending, and direct manipulation through parametric spatial anchors. These techniques make spatial evidence available at different locations and times while giving people concrete ways to specify, inspect, and revise what an AI system produces.
3. **A reusable benchmark for capability-conditioned spatial reasoning.** CapNav formalizes alignment as a navigation problem in which route feasibility depends on a specified human or robotic embodiment. Its videos, navigation graphs, capability profiles, edge-level traversability annotations, and decomposed metrics distinguish a plausible answer from a valid path, a traversable route, and an appropriate explanation. The benchmark reveals where current vision-language models fail under metric, multi-view, and embodiment-specific constraints.
4. **Empirical evidence and design knowledge from real spatial tasks.** Formative studies, user studies, expert reviews, and technical evaluations examine the systems with accessibility stakeholders, residents, building managers, occupants, AR creators, and designers, as well as across diverse indoor scenes and images. These studies establish where the proposed techniques are feasible and useful, while identifying limitations in personalization, capture coverage, latency, reconstruction, generation quality, and model reliability. Collectively, the findings show why imperfect AI can still support spatial work when its evidence and outputs remain inspectable, correctable, and bounded to an appro-

priate task.

1.6 Chapter Overview

Chapter 2 develops the conceptual and technical foundation for the dissertation. It first reviews human spatial perception as embodied and action-oriented, partial and individually variable, and organized around task-relevant objects, landmarks, and affordances. It then traces AI world understanding from integrated robotics and geometric mapping to learned visual geometry and multimodal embodied models. The chapter situates the dissertation among established human-AI interaction frameworks, reviews accessible navigation and environmental auditing as a central application strand, and identifies the gaps that motivate the four frontiers.

The four subsequent frontier chapters form the empirical core of the dissertation. *Frontier I: In-Context, Real-Time Personalized Spatial Assistance* presents RASSAR. *Frontier II: Spatial Perception Beyond Physical Presence* presents RAIS and FlyMeThrough. *Frontier III: Aligning AI Spatial Reasoning with Human Needs* presents CapNav. *Frontier IV: Human-Centered Spatial Representation and Interaction* presents SonifyAR and DepthScape. Each chapter begins with a preamble defining the frontier, then develops its example project or projects through their motivation, design, implementation, evaluation, and implications for human-AI symbiotic spatial perception.

The synthesis chapter draws lessons across the four frontiers and outlines a future in which AI can extend human spatial perception without displacing human judgment. The conclusion then returns to the central thesis and its implications. The goal is not to make physical presence universally unnecessary, nor to treat AI perception as complete. It is to make spatial assistance more timely, far-reaching, capability-aligned, and expressive, while giving people meaningful ways to understand and shape the representations on which that assistance depends.

Chapter 2

Overview of Human and AI Spatial Perception

This chapter establishes the conceptual and technical background for *human-AI symbiotic spatial perception*. It first reviews complementary theories of embodied perception, action, spatial knowledge, and mediated understanding. It then traces AI world understanding from geometric mapping and learned visual representations to multimodal embodied models. Next, it situates this dissertation among established human-AI interaction frameworks. Finally, it reviews accessible navigation, environmental auditing, and other spatial-assistance systems and develops the gaps that motivate the four frontiers.

2.1 Human Spatial Perception

Every animal must acquire useful information about its surroundings and act on that information to survive. In humans, visual depth and motion cues, hearing, touch, proprioception, vestibular sensation, and memory together support this basic capacity. People use these capabilities to infer layout, locate objects and surfaces, anticipate routes, and select actions. Human spatial perception is therefore not a passive recording of complete geometry. It is an ongoing process of sensing, remembering, interpreting, and acting on an environment under the constraints of a particular body, task, and situation.

2.1.1 Perception and Action in an Embodied World

Biologists and psychologists have long studied this relationship between an organism and its surroundings. Von Uexküll’s concept of *Umwelt* describes the meaningful world available to an organism through its particular perceptual and action capabilities [Uexküll, 2010]. An environment may be physically shared yet differently available to bodies with different senses, sizes, skills, and needs. A stairway, doorway, audible signal, reflective surface, or visual landmark therefore does not carry the same practical meaning for every person or machine. This perspective shifts spatial computing away from asking only what is present and toward asking what a particular actor can perceive and do.

Ecological theories make this relation to action explicit. Gibson’s affordances describe opportunities for action relative to an organism and its environment [Gibson, 1979]. A chair affords sitting, a handle affords grasping, and an opening affords passage under particular bodily and situational conditions. Norman’s distinction between affordances and signifiers adds an interaction-design perspective: an opportunity for action may exist, but a perceivable cue is needed to communicate it [Norman, 2013, 1999]. For spatial perception, simply detecting an object or space is therefore only an intermediate result. A useful system must relate the physical presence to possible actions and communicate that interpretation without concealing uncertainty.

Our perception is also active. Bajcsy argues that perceptual systems strategically control sensing in response to goals rather than waiting for complete input [Bajcsy, 1988]. People turn their heads, approach an object, open a door, or revisit a viewpoint to resolve uncertainty. Recent experimental work further shows that motor engagement during exploration can improve route memory compared with passively viewing the same route, emphasizing that spatial knowledge depends on how an environment is experienced as well as what is seen [Sivashankar et al., 2025].

2.1.2 Organizing Spatial Knowledge

Spatial-cognition research explains how people organize the evidence collected through perception and action. Tolman’s cognitive-map work introduced internal spatial representation as an explanation for flexible navigation [Tolman, 1948]. Lynch showed how paths, edges, districts, nodes, and landmarks contribute to the imageability and legibility of cities [Lynch, 1960], while Siegel and White described the acquisition

of landmark, route, and survey knowledge [Siegel and White, 1975]. These accounts emphasize different aspects of spatial knowledge, but all reject the idea that people maintain a complete, perfectly metric survey map.

Tversky further characterized spatial knowledge through cognitive maps, cognitive collages, and spatial mental models that are partial, schematic, and shaped by the current task [Tversky, 1993]. Montello showed that object-scale, vista-scale, environmental-scale, and geographic-scale spaces are apprehended and used differently [Montello, 1993]. Taylor and Tversky, meanwhile, demonstrated that descriptions and maps preserve different relations and support different forms of spatial inference [Taylor and Tversky, 1992]. These theories are complementary rather than interchangeable. Affordances explain how spatial features relate to possible action; active perception explains how evidence is sought; cognitive-map and landmark theories explain how locations and routes are organized; and scale and representation theories explain why the same environment may be understood differently across tasks and media. No single construct serves as the target of this dissertation. Together, the theories establish that spatial understanding is selective, action-oriented, and assembled from multiple forms of evidence.

2.1.3 Sources, Scales, and Limits of Spatial Knowledge

Across these theories, useful spatial knowledge draws on three complementary sources of information:

1. *Immediate sensory information* comes from the current situation: visual depth and motion cues, sound, touch, bodily orientation, and active inspection. This source is grounded in the present viewpoint but is necessarily limited by occlusion, attention, and sensing range.
2. *Retrieved knowledge* comes from memory and prior experience. It lets people recognize landmarks, remember a route, infer likely object locations, and relate a current view to a previously visited place. It is useful precisely because direct sensing is incomplete, but it can be outdated, distorted, or unavailable for unfamiliar environments.
3. *Mediated representations* come from other people and external media, including language, sketches, maps, photographs, videos, and online information. Such representations can make remote or past information available, but their value depends on what they preserve and how well a person can inte-

grate them into the current task. Past research shows that people acquire different spatial knowledge from maps, route descriptions, virtual environments, and direct navigation [Tversky and Lee, 1999; Richardson et al., 1999].

These categories are analytical rather than mutually exclusive. A person reading a map may combine it with remembered landmarks and current visual cues; someone who asks for directions must connect another person’s verbal description with an immediate view. The important point is that spatial understanding is assembled from heterogeneous evidence, not copied from a single sensor or data structure.

Spatial knowledge also remains partial because people are bounded by their position in space and time. Time-geography characterizes reachable activities through capability constraints, coupling constraints that require coordination with other people or resources, and authority constraints that regulate access to places [Hägerstrand, 1970]. Physical travel consumes time and may be risky, expensive, socially constrained, or impossible. The resulting *epistemic horizon* is the practical boundary of what a person can currently know about an environment. It changes with viewpoint, attention, prior experience, mobility, and access to other people or technologies.

Technology can extend this horizon through remote sensing, recorded media, mapping, or communication with others, but it cannot make every environment fully known. A person standing in a room can inspect visible surfaces but not space behind a wall; a video can reveal a distant room while omitting scale, sound, or objects outside the camera path. The design problem is therefore not to construct an exhaustive representation of every environment. It is to identify which missing evidence constrains a decision, acquire or infer that evidence responsibly, and express it in a form that supports the person’s current understanding and action.

2.2 AI World Understanding

AI world understanding refers here to computational methods for recognizing physical entities, estimating spatial structure and state, integrating evidence over time, predicting change, and selecting actions. These capabilities have not developed along a single path. Symbolic and geometric representations remain important, while learned visual, multimodal, and action-conditioned representations have expanded the range of environments and tasks that machines can address.

2.2.1 From Geometric Mapping to Learned Scene Representations

Early integrated AI and robotics systems made the perception-representation-action loop concrete. Shakey combined sensing, an explicit representation of its surroundings, planning, and mobile action [Nilsson, 1984]. Subsequent robotics research developed increasingly capable methods for mapping, localization, obstacle avoidance, and motion planning. Simultaneous localization and mapping, for example, estimates an agent’s trajectory while constructing a map from sequential observations, making metric structure and uncertainty explicit [Cadena et al., 2016; Mur-Artal and Tardós, 2017]. Such systems provide inspectable geometric state, although they can depend on sensor assumptions, engineered correspondence, and environment-specific failure handling.

The large-scale deep-learning era shifted emphasis toward representations learned from data. ImageNet supplied a large hierarchical image database and benchmark for visual recognition [Deng et al., 2009], and AlexNet demonstrated the impact of deep convolutional networks and GPU training at that scale [Krizhevsky et al., 2012]. Deep vision subsequently improved object detection, segmentation, depth estimation, and scene understanding. Matterport3D added RGB-D observations and reconstructed indoor environments [Chang et al., 2017], while Habitat enabled embodied agents to be trained and evaluated across many scanned spaces [Savva et al., 2019].

Learned visual geometry is now beginning to unify tasks that were previously handled by separate pipelines. VGGT, for example, predicts camera parameters, depth maps, point maps, and 3D point tracks from one to hundreds of views in a single feed-forward model [Wang et al., 2025a]. Scene graphs organize objects, attributes, and relations [Johnson et al., 2015]; CLIP provides transferable image-language representations [Radford et al., 2021a]; and SAM 2 supports promptable segmentation and propagation across images and video [Ravi et al., 2024a]. Together, these methods expose increasingly rich geometric and semantic evidence that can be assembled into spatial applications.

2.2.2 Multimodal and Embodied Spatial Intelligence

Multimodal foundation models connect images, video, language, and other sensor streams through more unified interfaces. Vision-language models can answer questions about scenes, generate descriptions, and ground language in visual evidence. PaLM-E interleaves continuous sensor inputs with text in an embodied

language-model architecture [Driess et al., 2023]. SayCan combines language-model task proposals with robot-specific affordance values [Ahn et al., 2022], while RT-2 represents robot actions as tokens and co-trains on robot trajectories and web-scale vision-language data [Zitkovich et al., 2023]. These systems move beyond recognition toward choosing actions in response to human instructions.

Recent robotics work has made spatial representation itself an explicit design target. SpatialVLA injects egocentric 3D position information and uses spatial action grids to support cross-robot manipulation policies [Qu et al., 2025]. Spatial MLLM connects a feed-forward visual geometry foundation model to a LLM backbone to enable spatial inference [Wu et al., 2025]. Such work illustrates a broader shift from asking whether language and images can be combined to asking what spatial structure must be preserved for reliable action. It also shows that semantic and spatial inference are related but distinct capabilities: recognizing an object or interpreting an instruction does not by itself determine a metrically valid motion.

Embodied models are therefore best understood as compositions of perception, representation, planning, and control choices. Some produce high-level plans that another controller executes; others predict discrete or continuous actions directly. Their effectiveness depends on embodiment, calibration, latency, safety constraints, and the ability to recover when observations or plans are wrong. These requirements are stricter than those of image-based question answering because errors have consequences in the physical environment.

2.2.3 Predictive World Models

The focus of this dissertation is distinct from *world model*, which is a learned predictive model of environment state and dynamics. Given observations and, often, candidate actions, it estimates how future observations or latent states may evolve. This definition follows the model-based-control use of the term rather than treating every spatial representation as a world model. A predictive world model can support planning by allowing an agent to compare possible futures before acting [Ha and Schmidhuber, 2018].

World models vary in what they predict and how directly that prediction corresponds to physical state. Pathdreamer predicts future visual observations along indoor navigation paths [Koh et al., 2021]. Navigation World Models scales action-conditioned video prediction across egocentric human and robot data and uses generated futures to plan or rank navigation trajectories [Bar et al., 2025]. More recently, Chain of World

couples a vision-language-action model with a latent representation of motion so that temporal prediction can inform robot action without reconstructing every background pixel [Yang et al., 2026]. These examples show how predictive representations can connect perception, temporal reasoning, and control.

Prediction introduces its own representational risks. A generated future may be visually plausible while violating metric geometry, object permanence, collision constraints, or the agent’s actual dynamics. Autoregressive rollouts can accumulate error, and a model trained on typical scenes can treat a statistically likely event as physically possible in the present scene. Predictive quality must therefore be evaluated against the downstream action and not only through visual realism. For human-facing assistance, systems must also communicate when a proposed future is inferred rather than directly observed.

2.2.4 Limits in AI Spatial Grounding

Current AI systems encode space as maps, meshes, point clouds, depth fields, scene graphs, navigation graphs, learned embeddings, token sequences, and generated media. Each representation supports some operations while discarding information needed by others. The central difficulty here is building a persistent spatial representation that can incorporate spatial relations and approximate scales based on the input evidence. A model may correctly identify object semantics while confusing their spatial relations, failing to combine observations across views, or giving incompatible answers about distance, size, relative position, or affordance. SpatialRGPT and Mind the Gap make these forms of reasoning explicit [Cheng et al., 2024; Stogiannidis et al., 2025]; VSI-Bench further shows that fluent video descriptions do not guarantee reliable metric or multi-view spatial judgments [Yang et al., 2025b].

These failures become more consequential in open, dynamic, or long-horizon tasks. OSI-Bench finds that spatial abilities developed in structured indoor settings do not reliably transfer to outdoor scenes with larger scales, changing layouts, and less constrained viewpoints [Wu et al., 2026]. Under incomplete evidence, vision-language models may also invent objects, relations, or explanations rather than signal uncertainty [Li et al., 2023; Guan et al., 2024]. Geometry-aware checks can improve particular tasks. For example, Context-Nav validates target descriptions with viewpoint-aware 3D relations during instance navigation [Jang and Kim, 2026]. However, these checks do not establish whether the available evidence is sufficient for a person’s question or whether the resulting judgment is valid for that person’s capabilities.

For human-facing spatial assistance, these weaknesses can turn a plausible answer into a misleading or risky one.

2.3 Human-AI Interaction Frameworks

Human-AI symbiotic spatial perception belongs to a longer history of frameworks for designing relationships between people and computational systems. These frameworks differ in scope and unit of analysis. Some describe a broad division of intellectual labor, some prescribe how control should move between a person and a system, and others provide concrete guidance for adaptation, error recovery, and appropriate reliance. Together, they establish that successful human-AI interaction cannot be inferred from model accuracy alone.

Licklider’s account of man-computer symbiosis proposed a tightly coupled partnership based on complementary strengths: people contribute goals, judgment, flexible reasoning, and problem formulation, while computers contribute rapid calculation, storage, and routine processing [Licklider, 1960]. The term *symbiotic* in this dissertation follows that commitment to complementarity rather than implying that a person and an AI system have equal agency or interchangeable perception. The spatial setting also makes the division of labor concrete. A person can identify a meaningful destination, accessibility concern, intent, and actually conduct any physical action; a computational system can gather online information, estimate physical measurements, reconstruct spatial geometry, or search for candidate routes. Neither contribution is sufficient by itself.

Mixed-initiative interaction makes this partnership operational by allowing the user or the system to take initiative as goals, uncertainty, and expected costs change [Horvitz, 1999]. This perspective is especially relevant when spatial evidence is incomplete. An AI system may guide a person toward an unobserved doorway, request another camera view, or propose a route; the person may supply a missing constraint, reject an irrelevant detection, or redirect capture. The important design question is therefore not whether the system is autonomous, but when initiative should shift and whether the person can understand and reverse that shift. The projects in this dissertation implement multiple forms of this exchange through scan guidance, correction interfaces, sparse annotation, editable anchors, and capability-conditioned queries.

Ability-Based Design supplies a complementary accessibility framework. It centers what people can do,

places responsibility for poor user-system fit on the system rather than the person, and calls for interfaces that adapt to users' abilities while making adaptations transparent and sensitive to performance and context [Wobbrock et al., 2011]. Human-AI symbiotic spatial perception aligns most directly with its accountability, adaptation, transparency, and context principles. RASSAR's customizable rubrics and CapNav's explicit capability profiles, for example, reject a single assumed body as the universal basis for spatial judgment. Elaborated in six projects and grounded spatial tasks, this dissertation extends Ability-Based Design's concerns from interface access to spatial perception and action.

More recent human-centered AI guidance addresses the probabilistic and evolving behavior of AI systems. The Guidelines for Human-AI Interaction organize recommendations around initial expectations, ordinary interaction, AI failure, and change over time; among other concerns, they emphasize timing services to context, showing relevant information, supporting correction, and explaining system behavior [Amershi et al., 2019]. Shneiderman's human-centered AI framework similarly argues that high levels of automation can coexist with high levels of meaningful human control, with reliability, safety, and trustworthiness treated as design requirements rather than assumed consequences of technical progress [Shneiderman, 2020]. These principles recur throughout this dissertation in anchored findings, previews, deletion and refinement controls, decomposed benchmark metrics, and representations that expose the basis of an AI output.

Trust-in-automation research sharpens the point. The appropriate goal is not maximal trust, but reliance calibrated to what a system can do in the present context [Lee and See, 2004]. Overreliance is dangerous when an AI system confidently misses a stair, underestimates a clearance, or hallucinates a connection between rooms; blanket distrust also discards useful measurements and computational reach. Spatial interfaces can support appropriate reliance by exposing provenance, coverage, uncertainty, and task boundaries, and by letting people inspect the evidence behind a conclusion or obtain more evidence when needed.

Human-AI symbiotic spatial perception complements these aforementioned frameworks rather than replacing them. Its distinct contribution is to specialize the existing principles towards spatial data and physically situated action. The four frontiers in this dissertation identify where this spatial specialization creates design work: output must arrive in context and at the pace of action; evidence must be acquired across relevant places, times, and viewpoints; judgments must be conditioned on the capabilities and goals of the person who will act; and machine spatial representations must be translated into forms people can perceive,

manipulate, and repair.

2.4 Combining AI World Understanding with Human Needs

The preceding frameworks become concrete in systems that augment immediate sensory evidence, retrieve information gathered in the past, or mediate knowledge about remote environments. Accessibility is a central strand of this work because it makes person-environment fit, representation, and the consequences of error unusually visible. It also has a long history that predates current AI models and should not be reduced to object detection or route generation.

2.4.1 Accessible Navigation and Nonvisual Spatial Cognition

Orientation and mobility (O&M) provides a foundational account of travel by people who are blind or have low vision. Orientation involves establishing and maintaining one's position and direction relative to the environment, while mobility concerns moving safely and effectively through it. O&M practice combines sensory information, landmarks, route structure, mobility aids, active exploration, and instruction rather than treating navigation as the delivery of turn commands alone [Long and Giudice, 2010]. This distinction matters for AI systems: reaching a destination is one outcome, but maintaining awareness, confidence, and the ability to make independent decisions can be equally important.

Research on blind spatial cognition likewise cautions against treating vision as a prerequisite for spatial knowledge. Giudice argues that the central issue is access to spatial information and the form in which it is encoded, because audition, touch, proprioception, language, and movement can all support spatial learning [Giudice, 2018]. Experimental work further shows that dynamically updated verbal descriptions of local geometry can support learning and wayfinding in a large indoor layout, although that study used blindfolded sighted participants and should not be read as a substitute for evaluation with blind travelers [Giudice et al., 2007]. For interface design, the implication is not that one modality is universally equivalent to another, but that nonvisual representations should be evaluated for the spatial relations, cognitive effort, and actions they support.

Several navigation systems embody different choices along this design space. NavCog uses smartphone sensing and installed Bluetooth beacons to provide turn-by-turn indoor guidance, nearby points of interest,

and warnings about features such as stairs; its original deployment was evaluated with blind participants in complex indoor and outdoor areas [Ahmetovic et al., 2016]. Microsoft Soundscape instead uses spatialized audio callouts and auditory beacons to support ambient awareness and user-directed exploration. Research on Soundscape found that engagement with spatial audio was associated with continued use among thousands of blind and low-vision users, while a separate study with sighted participants found that auditory-beacon navigation supported greater spatial learning than conventional turn-by-turn guidance [Liu et al., 2022; Clemenson et al., 2021]. Clew offers another model: people can record and later replay an indoor route through audio and haptic guidance without requiring installed building infrastructure [Yoon et al., 2019].

Assistive AR and live description systems move beyond route instructions toward information anchored in the current scene. Zhao et al. designed AR visualizations that emphasize stair boundaries for people with low vision [Zhao et al., 2019]. WorldScribe combines perceptual models to produce live, context-aware visual descriptions while negotiating a tradeoff between descriptive richness and latency [Chang et al., 2024]. Beyond Omakase shows that blind travelers using navigation robots may prefer different levels of shared control across social situations, including the ability to monitor the environment and intervene in navigation decisions [Kamikubo et al., 2025]. Taken together, this literature shows that accessible navigation is not only a localization problem. It concerns the traveler’s ongoing awareness, agency, preferred modality, and authority to question or redirect assistance.

2.4.2 Accessibility Mapping and Environmental Auditing

Navigation assistance depends on information about whether environments and routes are usable. Accessibility, however, is not a fixed label attached to a place. It emerges from the relationship among environmental features, a person’s abilities and mobility technology, and the activity being attempted. Traditional home assessments therefore combine physical measurements with professional or lived expertise, often through checklists used by residents, caregivers, and occupational therapists [Clemson, 1997; Iwarsson, 1999; Law et al., 1996]. Such methods provide depth and accountability but can require an in-person visit, specialized knowledge, and substantial effort.

Researchers have explored how remote imagery, crowdsourcing, and computer vision can expand the

coverage of accessibility data. Hara et al. showed that crowd workers could identify sidewalk accessibility problems in Google Street View imagery, and Project Sidewalk developed this approach into a web-based tool for collecting labels at urban scale [Hara et al., 2013; Saha et al., 2019]. Subsequent work used these labels to train models for automatically detecting sidewalk barriers [Weld et al., 2019]. Indoor and building-scale work has extracted accessibility information from floor plans, point clouds, and geometric networks, including passable space, inaccessible entrances, and route connectivity [Kostic and Scheider, 2015; Balado et al., 2017; Dao and Thill, 2018]. These approaches reduce collection cost and can reveal patterns unavailable through isolated audits, but their results remain bounded by image coverage, reconstruction quality, labeling assumptions, regulations, and the represented mobility profile.

This literature exposes a gap between scalable auditing and individualized use. A standards-based label can support screening without determining whether a feature works for a particular person. A crowd or model may identify a curb ramp while missing surface condition, temporary obstruction, reachability, or the exact measurement needed for a decision. Conversely, requiring every person to visit and measure every location negates the value of remote spatial assistance. RASSAR and RAIS investigate this gap through situated and robotic indoor auditing; FlyMeThrough studies a more general human-guided infrastructure for preserving locally meaningful facilities; and CapNav tests whether model judgments remain valid when the represented embodiment changes.

2.4.3 Situated and Remote Spatial Assistance

Spatial technologies can also support inspection and decision making before a person is physically present. Embodied Exploration enables wheelchair users to inspect high-fidelity virtual replicas through an embodied avatar before visiting a location [Pei et al., 2023]. Accessibility Scout uses an LLM-based photo-scanning workflow to identify accessibility concerns while allowing people to tailor scans to their mobility, preferences, and particular environmental interests [Huang et al., 2025]. SceneScout makes street-level imagery available for route preview and virtual exploration by blind users, while revealing that plausible description errors can be difficult to detect without visual verification [Jain et al., 2026]. Advances in geometric reconstruction and real-to-simulation conversion can broaden the evidence available for such systems [Wang et al., 2025a; Xie et al., 2025], but they cannot independently decide what evidence matters or when it is

sufficient for a specific person’s judgment.

At present, physical-world assistance remains far less integrated into everyday life than digital AI assistance. Reliable support requires timely, in-context evidence; spatial knowledge that reaches beyond a person’s immediate presence; reasoning aligned with individual capabilities and goals; and interaction techniques that let people inspect, communicate with, and revise a system’s representation. Together, these requirements define the four research frontiers of this dissertation.

Chapter 3

Frontier I: In-Context, Real-Time Personalized Spatial Assistance

3.1 Preamble

The first research frontier asks how people and AI systems can perceive a spatial situation together while it is unfolding. Human spatial understanding is continuously updated through movement, attention, and action; support is useful only when it connects to the environment a person is currently experiencing and arrives in time to influence a decision. As defined in Chapter 1, spatial assistance is *in context* when it is grounded in the present scene, *real time* when end-to-end feedback arrives within the decision horizon of an ongoing action, and *personalized* when its interpretation reflects the capabilities, goals, and preferences of the person using it. For RASSAR, real time refers to incremental reconstruction, scan guidance, and accessibility findings that can alter where a user points the phone or what the user verifies during a scan; it does not imply that the complete room audit finishes within a subsecond control loop.

This frontier matters because many consequential spatial judgments are made in the middle of activity: whether a route is clear, a surface is reachable, an object poses a hazard, or a room can support a planned task. Accessibility makes the need especially visible. The same physical feature can be irrelevant, helpful, or prohibitive depending on a person’s mobility, assistive technology, routines, and immediate purpose. Recent systems for live visual description and shared-control navigation likewise show that useful assistance

depends on timing, agency, and the opportunity to intervene, not only on perceptual accuracy [Chang et al., 2024; Kamikubo et al., 2025].

Bringing these requirements together remains difficult. A system cannot assess an unseen feature, yet a user may not know which viewpoint or measurement the system needs. Object recognition alone does not establish whether a cabinet, threshold, or corridor is accessible or safe. Models also remain fallible, so users must be able to inspect a finding in place, contribute additional evidence, and correct it when necessary. Finally, fixed standards provide useful baselines but cannot fully capture the variation among bodies, homes, activities, and preferences; personalized accessibility-scanning research makes this variation central to the assessment process [Huang et al., 2025].

The resulting design challenge is to make sensing, interpretation, and human judgment part of one accountable loop: guide evidence collection, relate geometric and semantic observations to user-relevant criteria, anchor feedback to the observed scene, and preserve a representation that supports later review. The RASSAR project examines this challenge through indoor accessibility and safety auditing.

3.2 RASSAR: Room Accessibility and Safety Scanning in Augmented Reality

RASSAR operationalizes this frontier as a mobile augmented-reality system that helps people inspect indoor spaces while they are physically present. The system combines phone-based LiDAR reconstruction, real-time object detection, extensible accessibility rubrics, in-situ visual and audio feedback, and a post-scan 3D summary. Its design and evaluation address the complete interaction loop: what stakeholders need from an audit, how a user collects evidence, how the system turns detections into accessibility findings, how those findings are communicated, and where human verification remains necessary.

3.2.1 Introduction

Safe and accessible living spaces are a fundamental human right [Organization, 2018]. Yet inaccessible housing remains prevalent throughout the world. In the United States, for example, 90% of housing units are inaccessible to people with disabilities [Smith et al., 2008; Steinfeld et al., 1998]. In the United King-

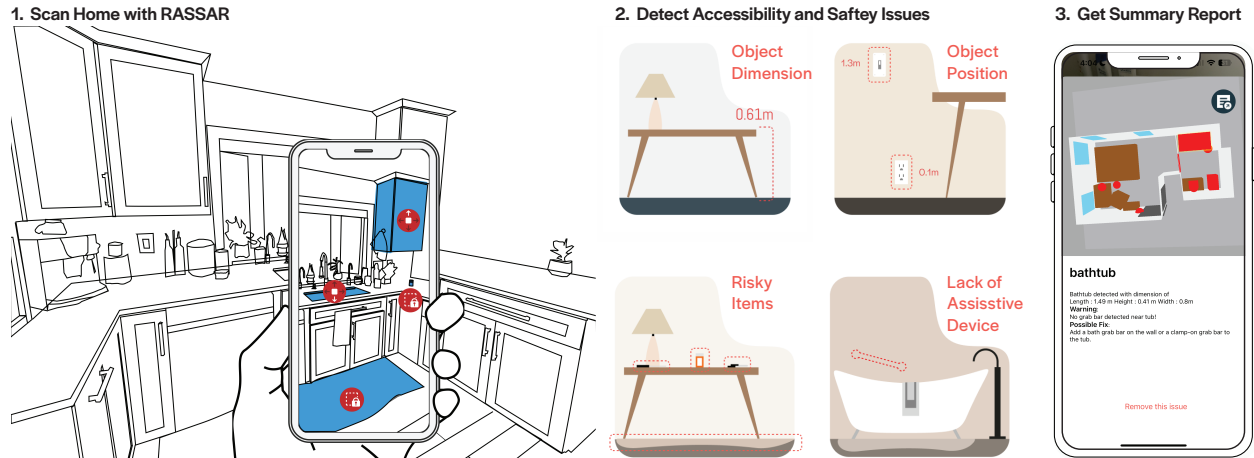


Figure 3.1: RASSAR is a mobile AR application for semi-automatically identifying, localizing, and visualizing indoor accessibility and safety issues. It scans a home using LiDAR and computer vision, evaluates detected elements against accessibility and safety rubrics, and presents an interactive 3D summary.

dom, 98% of newly built private homes are inaccessible to wheelchair users [Imrie, 2003]. To improve the safety and accessibility of domestic spaces, researchers and health professionals have created preformatted checklists that help residents and trained professionals audit and renovate indoor spaces [Struckmeyer et al., 2021; Clemson, 1997; Iwarsson, 1999; Fänge and Iwarsson, 2005]. For example, the *Home Safety Self-Assessment Tool* (HSSAT) [Horowitz et al., 2016; Tomita et al., 2014] includes a checklist for issues such as uneven flooring, cluttered areas, slippery throw rugs, and inaccessible light switches across nine areas of the home (e.g., kitchens, bathrooms, and bedrooms). Other researchers have explored remote assessment methods using teleconferencing and video cameras [RENDAL and LAPE, 2018; Romero et al., 2017].

With advances in computer vision (CV), augmented reality (AR), and mobile sensors, new ways to assess indoor accessibility and safety are now possible. For example, emerging smartphones contain built-in Light Detection and Ranging (LiDAR) sensors [Apple, 2020], which can reconstruct indoor spaces in real time with high precision [Díaz Vilariño et al., 2022]; new CV models such as YOLO [Jocher et al., 2022] can recognize objects across categories; and smartphones contain powerful computational units for on-device processing. Leveraging these advances, we created RASSAR (Room Accessibility and Safety Scanning in Augmented Reality), a custom mobile AR application for semi-automatically identifying, localizing, and visualizing indoor accessibility and safety issues using LiDAR and real-time computer vision (Figure 3.1). With RASSAR, a user scans an indoor space with their phone; the tool constructs a real-time

parametric model of the 3D scene, identifies and classifies known accessibility and safety issues, and visualizes potential problems in AR overlays. Throughout the scan, users can confirm or delete automatic detections, or select them to view more information (Figure 3.5). After scan completion, RASSAR shows an interactive 3D room reconstruction with a summary of findings (Figures 3.1 and 3.6). To assist blind or low-vision users, RASSAR supports VoiceOver and provides custom audio assistance during scanning, including verbalizations of detected items.

To develop RASSAR, we conducted a three-stage iterative design process. First, informed by prior work [Tomita et al., 2014; Horowitz et al., 2016; , 2010; Housing and inc, 1996], we built a *rapid technical prototype* to demonstrate feasibility and examine initial audit possibilities. Second, using this prototype as a design probe [Saha et al., 2022; Jain et al., 2015; Froehlich et al., 2012; Hara et al., 2016], we conducted a *formative study* with 18 participants across five stakeholder groups: wheelchair users, blind or low-vision participants, families with young children, caregivers, and occupational therapists. We showed videos of RASSAR auditing an apartment, solicited feedback on new interface mockups, and engaged participants in brainstorming new features. The study revealed common challenges in home accessibility and auditing that RASSAR could potentially alleviate, along with suggested improvements to its rubrics and user interface. Third, we *built the current RASSAR system* with improved detection performance, user interaction, and interface design and performed two additional studies: a technical performance evaluation (Study 2) and a user study (Study 3) with six participants again drawn from these stakeholder groups.

As emphasized in prior literature [Struckmeyer et al., 2021] and further affirmed by our formative study, home audit tools must be adaptable to individual needs and differences across communities. To address this challenge, RASSAR introduces a custom, fully extensible JSON format for encoding specific accessibility and safety issues in its audit engine. RASSAR currently supports 20 issues across four problem categories: *object dimension* (e.g., a table that is too low), *object position* (e.g., a light switch that is too high), *risky item* (e.g., a throw rug), and *lack of assistive item* (e.g., no grab bar near a toilet). Users can add issues by editing the JSON specification directly. In the future, an authoring tool could make such additions more accessible.

To evaluate RASSAR’s performance and usability in real-world settings, we conducted the aforementioned technical evaluation and user study. In the technical evaluation, the first author visited ten homes of varying sizes and layouts, conducting a manual baseline audit and three RASSAR scans in each home.

Repeating each scan allowed us to examine consistency. The user study followed a similar procedure, but participants conducted the RASSAR scans independently. We then interviewed participants about their experiences to gather qualitative insights. Average precision and recall were 0.86 and 0.83, respectively, for scans conducted by the researcher and 0.79 and 0.73 for scans conducted by participants. Participants completed audits with RASSAR 3.5 times faster, on average, than the manual audits. As P4, a wheelchair user, said, “RASSAR is easy to use and was pretty accurate in terms of ADA.”

In summary, this project contributes (1) formative findings from five stakeholder groups about indoor accessibility and safety auditing, the reconfiguration of spaces to fit individual needs, and reactions to future tools that use advanced sensing; (2) RASSAR, an AR-based tool for semi-automatically detecting accessibility and safety issues in indoor spaces using LiDAR and real-time computer vision; and (3) findings from a technical evaluation and follow-up user study that characterize RASSAR’s performance, usability, and potential applications. The project also introduces an extensible JSON format for specifying indoor accessibility and safety issues, together with a custom object-detection model and its training dataset.

Our work aims to transform how people examine and configure indoor spaces to improve accessibility and safety. We envision RASSAR as a versatile tool that could help builders assess the safety and accessibility of new construction, residents plan renovations or update their homes in response to life changes (e.g., illness or childbirth), rental platforms and hosts communicate the accessibility and safety of rental spaces, and occupational therapists assist residents in identifying accessibility and safety issues during home visits.

3.2.2 Design Process

To design RASSAR, we conducted a three-stage iterative design process. We first built an *initial technical prototype* to demonstrate feasibility. We then used the prototype as a design probe in a *formative user study* with five key stakeholder groups (Study 1). Finally, informed by the formative study, we built the *current RASSAR system* with improved detection performance, rubric construction, user interaction, and interface design and performed both a technical evaluation (Study 2) and a user study (Study 3).

All RASSAR studies involving human participants were reviewed and approved by the University of Washington Institutional Review Board (UW IRB; STUDY00016035).

3.2.2.1 Technical Prototype

Informed by prior work in home accessibility assessment [Balado et al., 2017; Fu et al., 2020] and indoor reconstruction techniques [Yang et al., 2020; Apple, 2022], we built a rapid technical prototype on an iPhone 13 Pro Max using Apple’s RoomPlan API [Apple, 2022]. The prototype included three primary features: (1) reconstruction of indoor spaces and accessibility-related items, (2) detection of accessibility and safety issues in indoor scenes, and (3) AR-based visualization of and interaction with detected issues. We tested the prototype in a single apartment to demonstrate technical feasibility and determine suitable scanning conditions while varying room tidiness, lighting, and movement speed [Su et al., 2022].

Our initial findings suggested that scanning should be conducted at a moderate speed (approximately 0.5 meters per second) in a tidy, well-lit room. Under these conditions, the technical prototype’s detection recall reached 90%. The experiment also revealed deficiencies in object-detection performance and limitations in the user interface, which we addressed in the current RASSAR system (Section 3.2.3).

3.2.2.2 Study 1: Formative Study

To examine the potential of semi-automatic accessibility and safety scanning with mobile phones and to solicit feedback on our initial technical prototype, we conducted a three-part formative user study with 18 participants drawn from five communities: wheelchair users ($N=8$), families with young children ($N=3$), people who are blind or have low vision ($N=4$), older adults ($N=6$, including caregivers), and occupational therapists ($N=3$); see Table 3.1.

Participants. Participants were recruited via email, advertisements to accessibility organizations, and social media as well as through snowball sampling. Before participating, all individuals filled out a screener on demographics and relationship(s) to our target communities. To ensure a diverse sample, we recruited at least three individuals from each community who were either self-identified members or caregivers.

Procedure. We conducted a three-part qualitative study. Part 1 addressed current practices, and Parts 2 and 3 focused on our RASSAR design probe. In Part 1, we asked participants about their indoor accessibility and safety needs, their current practices for assessing those needs, and the challenges they encountered. In Part 2, we showed participants videos of RASSAR scanning an apartment and solicited suggestions,

Table 3.1: Demographics of the 18 formative-study participants. PID denotes participant ID; “Older adults” identifies participants aged 65 or older; “Children” identifies families with children from birth through age three; BLV denotes participants who are blind or have low vision; OT denotes occupational therapists; and CG denotes caregivers. Roles and identities are not mutually exclusive.

PID	Wheelchair User	Older Adults	Children	BLV	OT
P1	✓				
P2	✓				
P3	✓				
P4	✓				
P5	✓				
P6	✓				
P7			CG		
P8	✓	✓			
P9		CG	CG		
P10	✓	✓			
P11					✓
P12					✓
P13		✓		✓	
P14		✓		✓	
P15		✓		✓	
P16			CG		
P17					✓
P18				✓	

concerns, and expected usage scenarios. In Part 3, we showed RASSAR user-interface mockups as design probes and asked participants about their preferences for encoding new accessibility and safety issues into RASSAR (e.g., rubric customization), guiding user-conducted scans, and communicating scan results and errors. Sessions lasted 40 to 70 minutes ($M=52$ minutes), and all but one were conducted over Zoom. The lead author conducted all 18 sessions. We include the full interview protocol and design probe in the supplementary materials.

Data and Analysis. We recorded all sessions and transcribed the recordings using Rev [Rev.com]. For analysis, we conducted reflexive thematic coding [Braun and Clarke, 2019]. The first author, who also conducted the interviews, reviewed all transcripts to develop an initial codebook. The first four authors then discussed the codebook, iteratively resolved disagreements, and collaboratively developed a final version. The first author applied the finalized codebook to all transcripts. After coding, the first four authors met to discuss broader themes and findings.

Findings. We highlight four findings related to indoor accessibility and safety needs, current assessment practices and challenges, reactions to RASSAR, and the design probe results.

Overall reactions to RASSAR. Most participants ($N=16$) held favorable opinions about the RASSAR prototype because of its measurement and documentation features, its potential to help prepare a home for

visitors with accessibility needs, and its customizable set of accessibility issues. The two participants who held unfavorable or neutral opinions stated, “I don’t see the point since I can do these screenings by myself” and “I don’t know.”

Participants described a variety of use cases for RASSAR. Six participants mentioned how RASSAR could help people learn about physical spaces before visiting them. For example, P18, who was blind or had low vision, said, “I think it could really help people have more confidence when they go into a room that they’re not familiar with or a new space.” Another commonly raised use case was facilitating renovations or real-estate viewings. For example, P4, a wheelchair user, said, “I could even see this as being incredibly helpful to send to contractors and architects and designers.” Finally, participants were excited about ways RASSAR could support travel accommodations, as hotel or short-term-rental managers could use scan results to communicate the accessibility of their spaces.

People’s unique indoor accessibility needs. Our participants identified common sources of indoor safety and accessibility issues, such as stairs ($N=14$), doors ($N=9$), and floors ($N=9$), which also commonly appear in accessibility checklists [Tomita et al., 2014; Horowitz et al., 2016; Together]. However, concerns also differed across communities. For example, no wheelchair user mentioned hazardous items such as sharp furniture corners or knives, whereas participants from all other communities raised these concerns (blind or low vision, $N=3$; older adults, $N=4$; families with children, $N=2$; occupational therapists, $N=1$). This variation highlights the need to customize RASSAR for different abilities and needs.

Current practices and challenges. None of our participants with accessibility needs previously chose to use accessibility checklists when auditing indoor spaces. They instead relied on their own experiences and formed methods through practice. The most commonly used methods were exploring the space ($N=6$), creating mind maps of potential issues ($N=5$), and asking questions about the space ($N=4$). Caregivers typically put themselves in care receivers’ shoes ($N=2$) to better identify potential issues. Current practices also posed inherent challenges, including other people’s limited understanding of accessibility ($N=3$) and social awkwardness ($N=3$).

For example, P10 said, “People mean well, they really do. [But] until you actually have to live it or be exposed to it on a routine basis, you just really don’t understand [our accessibility challenges].” Similarly, P6 said, “The main challenge is explaining to people how to look at it from my perspective.” Conducting

an audit on site with the person whose needs are being assessed may improve its relevance, but doing so can require additional time or expense. P8 recalled calling a restaurant to confirm accessibility, only to find it inaccessible upon arrival and having to choose a nearby alternative. Indoor auditing methods such as RASSAR could mitigate these challenges by providing a digital representation of a space that people can evaluate remotely against general or personalized requirements.

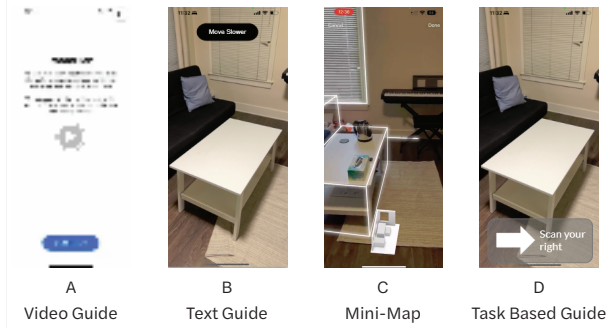
Design improvements for RASSAR rubrics and UI. We presented participants with a list of accessibility and safety issues derived from the literature and solicited their feedback. Most participants were satisfied with the list, while some suggested additions. Based on this feedback, we modified and finalized RASSAR’s auditing rubrics, for example by adding bed-height issues. We also provided user-interface mockups to elicit preferences for the RASSAR scanning experience (Figure 3.2). Participants preferred text hints ($N=15$) and mini-maps ($N=11$) for scanning support, an interactive rich-text pop-up layer ($N=13$) for visualizing detected issues, an interactive 3D model ($N=9$), and a list of issues to review ($N=10$) in the post-scan summary. For error reporting, most participants ($N=15$) preferred that the system learn their needs over time.

3.2.3 The RASSAR System

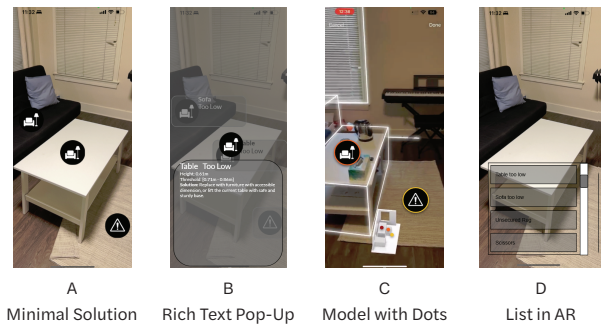
Informed by our rapid prototype and formative study findings, we created a revised RASSAR system (Figure 3.3) that uses LiDAR and real-time computer vision to reconstruct indoor spaces parametrically, identify safety and accessibility issues using a custom JSON rubric, and visualize issues in AR and through an interactive 3D summary view.

The new prototype includes four key enhancements. First, improved feedback presents detected accessibility and safety issues in real time through several interfaces, including AR pop-up icons, descriptive layers, and post hoc summaries in the 3D model. This feedback enables users to explore, refine, and summarize scan outcomes. Second, textual and minimap guidance reduces scanning effort by allowing users to scan spaces without approaching every potential object of interest. Third, users can select target accessibility communities to filter the auditing rubric and remove issues that do not match their needs. Fourth, to assist people who are blind or have low vision, the primary interactions for selecting target communities, scanning, identifying objects, and reviewing results provide audio feedback in real time. All interface components,

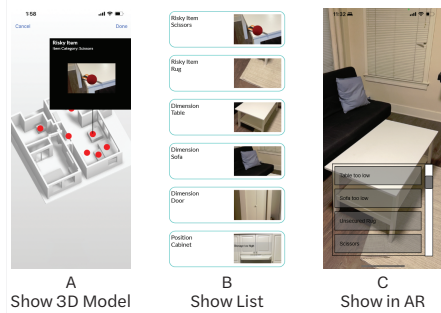
How should RASSAR support scans?



How Should RASSAR Show the Detected Issues?



How Should RASSAR Summarize Scan Results?



How Should Users Report Errors?

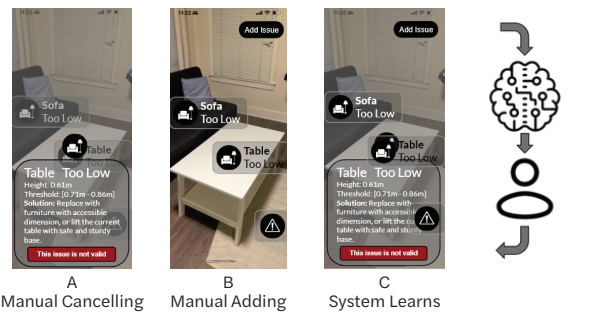


Figure 3.2: Design probes for RASSAR’s interaction and interface design. For each question, we provide three or four options as interface mock-ups.

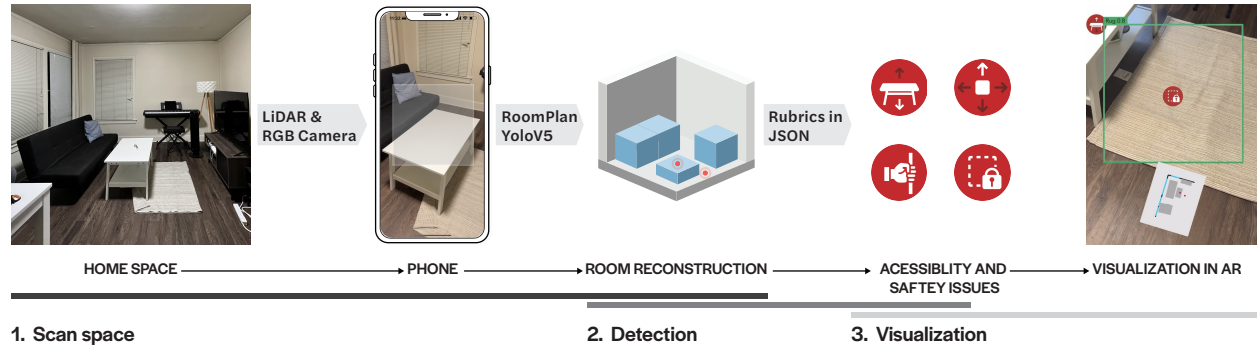


Figure 3.3: RASSAR system overview. RASSAR (1) scans and reconstructs a home space, (2) detects accessibility and safety issues in the space, and (3) visualizes the results in real time through AR.

such as buttons and text labels, are compatible with VoiceOver. Below, we describe RASSAR’s scanning, detection, and visualization process.

3.2.3.1 Scanning and Reconstructing a Home Space

In Step 1 of RASSAR’s technical pipeline, we reconstruct a scanned home space into a parametric 3D model that includes object category, dimension, and position information. To capture both large objects, such as furniture, and small objects, such as electric sockets, we combine Apple’s RoomPlan API [Apple, 2022] with a customized YOLOv5 model [Jocher et al., 2022].

Parametric Reconstruction of Major Indoor Components. The RoomPlan API is the backbone of our indoor reconstruction process. RoomPlan, which relies on both RGB camera and LiDAR sensor data, provides real-time spatial and dimension information for major indoor components. Currently, RoomPlan detects the following object categories: *bathtub, door, opening, wall, window, bed, chair, sink, sofa, stairs, storage, table, television, and toilet*. However, this reconstruction is incomplete; it includes only walls, doors and furniture but neglects smaller objects that could also pose accessibility and safety challenges.

Object Detection for Smaller Indoor Objects. Based on our literature review, we identified many issues related to smaller objects, such as grab bars, electrical outlets, and sharp objects. Because no existing pretrained model was tailored to accessibility-related indoor objects, we trained an object-detection model to complement the room-reconstruction pipeline. Informed by [, 2010; Housing and inc, 1996; Tomita et al., 2014; Horowitz et al., 2016; Kids Clinic, 2023; Nguyen et al., 2021], we selected nine initial categories of smaller indoor items that commonly create problems and collected a custom dataset with 2,533 images and 3,943 annotations (Table 3.2). Approximately one third of these images came from the Open Images dataset [Kuznetsova et al., 2020], while the remainder came from Microsoft Bing [Microsoft]. The first author manually annotated all instances of the nine object categories with bounding boxes.

Using the dataset, we trained the YOLOv5 computer-vision model [Jocher et al., 2022]. We selected YOLOv5-m (41 MB) because of its fast detection speed (224 ms CPU inference time [Ultralytics, Accessed September 11, 2023]) and good performance on the COCO benchmark (0.63 mAP@0.5¹). In offline exper-

¹mAP@0.5 is mean average precision at an intersection-over-union threshold of 0.5.

iments, a larger model slightly increased performance (0.66 mAP@0.5) but nearly doubled detection time (430 ms). We trained the model for 900 epochs on an NVIDIA GTX 3080 with Ubuntu 20.04.2 LTS. After obtaining the best weights, we converted them to Core ML format (.mlmodel) [Apple] and added a non-maximum-suppression layer [Jocher et al., 2022]. We randomly split the dataset into training, validation, and test sets containing 70%, 15%, and 15% of the data, respectively. To facilitate open science, we released the annotated dataset.²

Table 3.2: Numbers of images and annotations used to train RASSAR’s custom YOLO model. Because some images contained multiple objects, the annotation count exceeds the image count.

Object	Images	Annotations
Door handle	370	530
Electric socket	181	370
Grab bar	395	503
Knife	451	622
Medication	325	688
Rug	377	470
Scissors	226	270
Smoke alarm	176	191
Light switch	138	299
Total	2,533	3,943

During scanning, the custom YOLOv5 model runs continuously on the camera feed to detect the nine categories of accessibility-related objects. Because YOLO outputs 2D image bounding boxes rather than 3D room coordinates, we cast a ray [Apple, 2023c] from the center of each bounding box into physical space. To improve accuracy and reduce noise and outliers, we average results across multiple frames and apply filtering thresholds. A YOLO-detected object is considered valid after five ray-casting results have location offsets below 0.3 meters and detection-confidence scores above 0.65.

Combining the results of RoomPlan and YOLOv5, RASSAR produces a real-time indoor reconstruction that includes category, dimension, and position information for both larger barriers and smaller indoor objects. This reconstruction provides a basis for accessibility and safety auditing.

²<https://github.com/makeabilitylab/RASSAR>

3.2.3.2 Detection of Accessibility and Safety Issues

In Step 2 of RASSAR’s pipeline, we filter and identify relevant accessibility and safety issues based on the selected stakeholder group(s) and a customizable rubric.

Rubric Formation. RASSAR’s auditing rubric draws on the ADA Design Guidelines [, 2010], the Home Safety Self Assessment Tool (HSSAT) [Tomita et al., 2014; Horowitz et al., 2016], the US Fair Housing Act Design Manual [Housing and inc, 1996; Steinfeld et al., 1998], and other sources [Taira and Carlson, 2014; Osborne et al., 2008]. Creating the default rubrics for each stakeholder group was iterative and additionally informed by Study 1. The final rubrics include 20 issues in four categories (Figure 3.1, panel 2; Figure 3.4): object dimension (e.g., high table height), object position (e.g., an out-of-reach light switch), risky items (e.g., sharp objects such as scissors), and lack of assistive items (e.g., missing bathtub grab bars). When RASSAR starts, users can select one or more target communities, which load the relevant rubric.

Object Dimension Object Too Tall or Short		Object Position Object Too High or Low		Risky Item If Existent		Assistive Item If Non-Existent	
Bed Height	   	Cabinet Height	   	Rug	   	Grab Bar Near Toilet	   
Table Height	   	Sink Height	   	Scissors	   	Grab Bar Near Tub	   
Counter Height	   	Knob Height	   	Knife	   	Fire Alarm	   
Door Width	   	Door Handle Height	   	Medication	   		
Opening Width	   	Light Switch Height	   				
		Outlet Height	   				
		Grab Bar Height	   				
		Grab Bar Height	   				

Figure 3.4: Informed by literature and our formative study, RASSAR can detect 20 types of accessibility and safety issues across four categories: object dimension, object position, risky item, and lack of assistive item. Black icons indicate the accessibility communities for which each issue is relevant. Safety and accessibility needs are fluid and shaped not only by disability but also by fatigue and time of day, so individuals may not map exactly to these categories. The custom JSON-based rubric could support more precise individual specification in the future, for example through a custom authoring interface.

JSON-Encoded Rubrics. To facilitate automated screening of room reconstructions and individual customization, we transformed the original text-based rubrics into JavaScript Object Notation (JSON). Each JSON-formatted rubric includes the target object category, such as a table; the relevant user community, such as wheelchair users; a dependent object when a rubric involves multiple objects, such as a tub for the issue *No Grab Bar Near Tub*; and violation criteria, such as a dimension below 68 cm or a relative distance

greater than 70 cm. We also enriched the JSON files with warning messages, issue descriptions, suggestions, and information sources. This context helps users understand detected issues and remove invalid or irrelevant results.

3.2.3.3 Visualization of Scans

Finally, in Step 3, RASSAR provides a real-time 3D reconstruction to aid scanning, a visualization of identified issues in AR, and a post-hoc summary of scan results.

Facilitating User Scanning. The RASSAR interface employs several forms of feedback to facilitate scanning. First, a dynamic minimap represents room-reconstruction progress (Figure 3.5d). The minimap adapts to the user’s orientation and uses distinct visual cues: black lines represent walls, yellow lines represent doors, and an orange triangle indicates the user’s position and direction. Second, to improve the detection of small indoor objects from a distance, RASSAR sends only the central portion of the camera feed to the YOLO model. A dimmed white box indicates this region (Figure 3.5b). Third, RASSAR presents all object-detection results in YOLO-style bounding boxes (Figure 3.5c). Finally, users can enable audio feedback that guides the scan with instructions such as “Please point camera at top and bottom of wall to initialize,” “Please slow down,” and “Please step away from wall.” The system also announces major indoor objects, such as doors, windows, tables, and sofas, when it detects them.

AR Visualization of Detected Issues. We also overlay detections in real time using AR. RASSAR encodes the four classes of accessibility issues, object dimension, object position, risky item, and lack of assistive item (Figure 3.4), with corresponding pop-up icons. Users can select an icon to inspect detailed information (Figure 3.5, panel 2) and verify whether the issue is valid. If it is not, the user can remove it (Figure 3.5e). With audio support enabled, RASSAR also announces detected accessibility or safety issues, such as “Too narrow door opening is detected!,” “The bed is too high!,” and “No grab bar detected near toilet!”

Post-Scan Interactive 3D Summary. After a scan, RASSAR provides a brief summary of detected issues, including counts by category, that VoiceOver can read aloud. RASSAR then shows an interactive 3D model

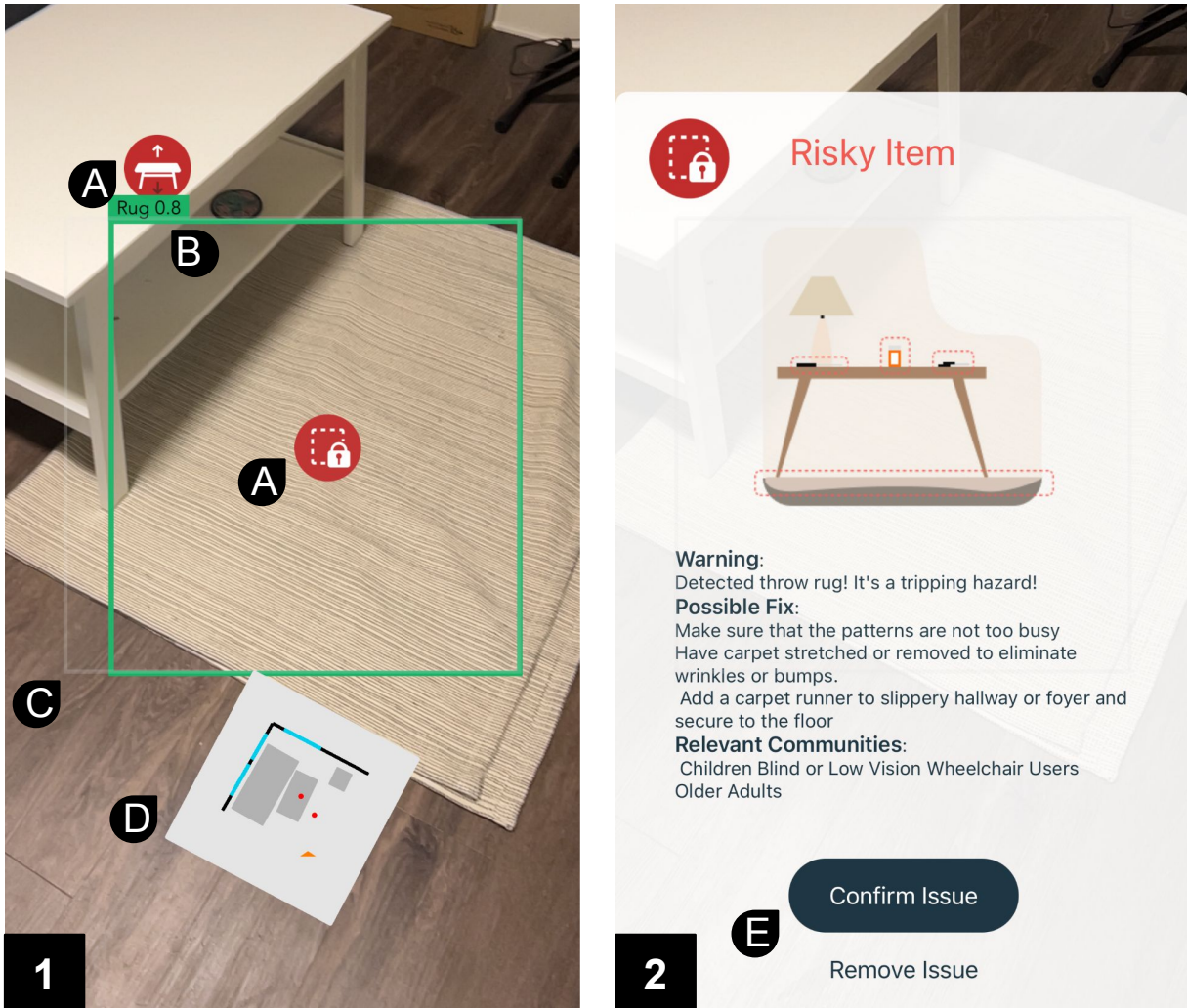


Figure 3.5: RASSAR's AR-based scanning interface and a detailed view of a detected issue. (1) During a scan, RASSAR shows detected problems in real time through AR overlays, including (a) red spheres that users can select to view more information and confirm or delete a detection, and (b) computer-vision detections with green bounding boxes, text labels, and confidence scores. A gray bounding box (c) indicates the computer-vision field of view. A minimap (d) adapts to the user's orientation and position and shows reconstruction results in real time. (2) The user can select an identified issue (a) to view more information and recommended solutions and can confirm or delete the issue (e).

of the results in which geometries represent indoor components and objects, and red marks indicate detected issues. The user can pan, zoom, and rotate the model for closer inspection or select an object to view more details. The lower half of the user interface shows the selected object's category and dimensions. The user can also delete detected issues or export all scan results to a JSON file.

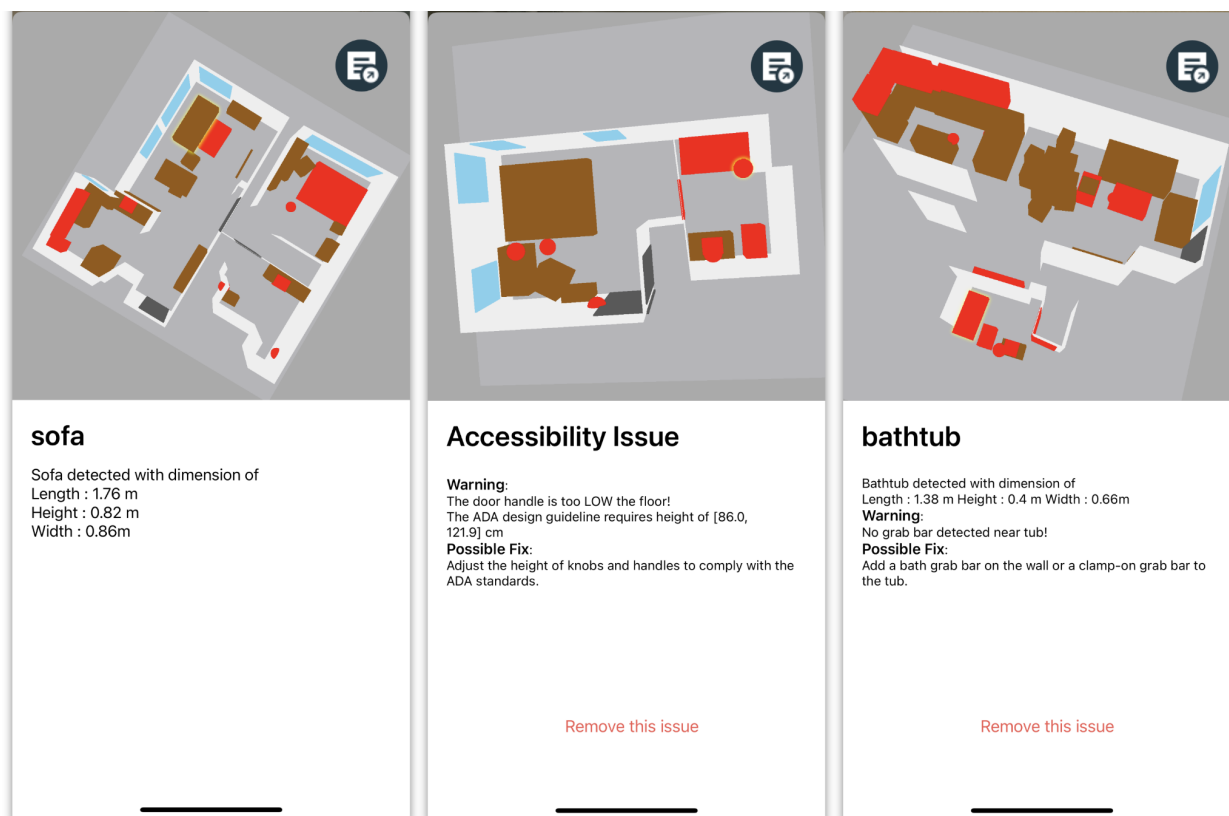


Figure 3.6: Three examples of RASSAR's post-scan interactive summary. A user can interact with the 3D reconstruction, inspect detailed information about objects or issues, and remove issues that are erroneous or do not apply. The button in the upper-right corner exports scan results to a JSON file.

3.2.4 Study 2: Technical Evaluation

Prior work in automatic accessibility auditing typically evaluates technical performance through case studies in three or four real-world environments [Balado et al., 2017; Deng et al., 2022; Díaz-Vilariño et al., 2019]. Expanding on this approach, we evaluated RASSAR in ten home spaces, including seven apartments and three houses of varying sizes and layouts (Table 3.3). The first author, an experienced accessibility researcher and RASSAR user, conducted all scans.

3.2.4.1 Experiment Procedure

For each indoor space, we performed a three-step process. First, we conducted a manual audit to identify existing accessibility and safety issues using RASSAR’s accessibility rubrics. We inspected and measured each indoor space, applied the rubric, and took additional notes about identified issues³. Second, we scanned each space with RASSAR and recorded and exported the results in JSON format. Finally, we compared the scan results with the manual-audit data. We repeated the final two steps three times, resulting in 30 RASSAR scans across ten indoor spaces.

3.2.4.2 Evaluation Metrics

To evaluate RASSAR, we used three primary evaluation measures: *detection performance*, *scanning consistency*, and *scanning time*.

Detection Performance. We calculated the number of true positives (TP), false positives (FP), and false negatives (FN) based on whether RASSAR successfully detected an issue listed in its accessibility rubric, reported an issue that did not apply, or failed to detect an applicable issue. We did not calculate true negatives because doing so would require enumerating accessible and safe items that RASSAR correctly omitted, which was not meaningful for this evaluation.

Based on TP , FP , and FN counts, we calculated four evaluation metrics: **precision** ($\frac{TP}{TP+FP}$), **recall** ($\frac{TP}{TP+FN}$), **F1 score** [Powers, 2020], and **accuracy** ($\frac{TP}{TP+FP+FN}$). Together, these metrics characterize RASSAR’s ability to capture applicable issues, avoid false alarms, and limit missed issues.

Scanning Consistency. To examine RASSAR’s consistency across scans (three per indoor space), we used Krippendorff’s alpha [Krippendorff, 2011], a common measure of agreement among multiple raters. We considered each scan a distinct rater and treated each accessibility and safety issue in a space as a scoring task.

Scanning Time. Finally, we computed the average scanning time for each indoor space using screen recordings of scans.

³We excluded the “Knob Height” issue from this study to avoid an imbalance in the manual-inspection results.

3.2.4.3 Evaluation Results

Table 3.3: Characteristics of the ten home spaces and RASSAR’s performance in each. Scanning time is reported in seconds, and each performance metric is averaged across three scans of the space.

ID	Home type	Size (m ²)	Rooms scanned	Average precision	Average recall	Average F1 score	Average accuracy	Krippendorff’s α	Scan time
S1	Apartment	65	3	0.75	0.91	0.82	0.70	0.73	113
S2	Apartment	63	2	0.72	0.73	0.72	0.56	0.7	120
S3	House	45	4	0.85	0.79	0.81	0.69	0.67	148
S4	Apartment	55	3	0.90	0.79	0.84	0.73	0.82	80
S5	Apartment	50	3	0.94	0.91	0.92	0.86	1	84
S6	Apartment	90	4	0.92	0.82	0.87	0.76	0.83	125
S7	Apartment	65	3	0.84	0.87	0.85	0.74	0.62	96
S8	Apartment	50	3	1.00	0.70	0.82	0.70	0.69	80
S9	House	24	2	0.73	0.90	0.81	0.68	-0.05	53
S10	House	60	3	0.92	0.88	0.90	0.82	0.43	100
Average				0.86	0.83	0.84	0.72	0.64	99.9

Table 3.3 and Figure 3.7 show the RASSAR evaluation results. The table presents average performance statistics for each space. Overall, RASSAR identified indoor accessibility and safety issues with a mean precision of 0.86, recall of 0.83, F1 score of 0.84, and accuracy of 0.72. Most Krippendorff’s alpha values fell between 0.4 and 0.7, indicating moderate to substantial consistency [Hughes, 2021]. RASSAR scans took 99.9 seconds on average ($SD=27.4$), compared with approximately 10 minutes per space for the lead author’s manual audits.

3.2.4.4 Performance Analysis

Below, we describe more detailed findings related to performance as a function of issue and potential error causes.

Performance by Accessibility and Safety Issue. RASSAR’s technical performance varies across accessibility and safety issues. Table 3.4 reports individual performance metrics for 13 commonly encountered issues. Most exhibit strong detection performance, with F1 scores exceeding 0.65. Bed height and cabinet height each achieved an accuracy of 1.0. The notable exception is counter height. Its rubric requires a counter surface to be 28 to 34 inches high, but the RoomPlan API consistently classified kitchen counters as storage units. This systematic error caused RASSAR to overlook counter-height issues in most scans.

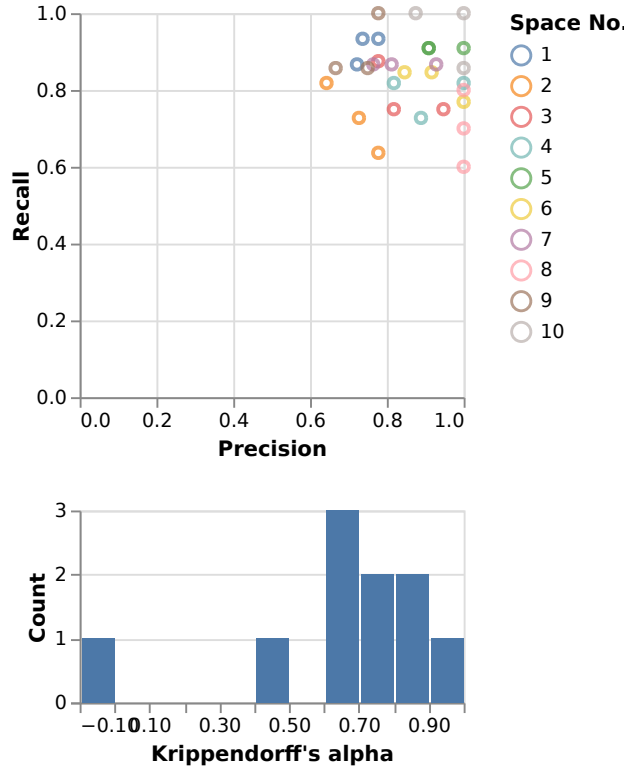


Figure 3.7: RASSAR’s technical evaluation performance over ten indoor spaces. (Top) A scatter plot of precision and recall over 30 scans in 10 home spaces. (Bottom) A histogram of Krippendorff’s alpha values across 10 home spaces (3 scans each).

Table 3.4: RASSAR detection performance by accessibility and safety issue. GT denotes the ground-truth count from manual audits; TP, FP, and FN denote true positives, false positives, and false negatives, respectively.

Category	Issue	GT	TP	FP	FN	Precision	Recall	F1 score	Accuracy
Object dimensions	Counter height	42	3	1	39	0.75	0.07	0.13	0.07
	table height	42	38	8	4	0.83	0.91	0.86	0.76
	Door radius	27	17	3	10	0.85	0.63	0.72	0.57
	Bed height	15	15	0	0	1.00	1.00	1.00	1.00
Object positions	Sink height	57	54	0	3	1.00	0.95	0.97	0.95
	Cabinet height	48	48	0	0	1.00	1.00	1.00	1.00
	Grab bar height	15	12	8	3	0.60	0.80	0.69	0.52
Presence of risky item	Rug	48	48	2	0	0.96	1.00	0.98	0.96
	Medication	21	19	18	2	0.51	0.91	0.66	0.49
	Knife	15	14	7	1	0.67	0.93	0.78	0.64
	Scissors	15	15	0	0	1.00	1.00	1.00	1.00
Absence of assistive item	No grab bar near toilet	27	24	0	3	1.00	0.89	0.94	0.89
	No grab bar near tub	18	16	0	2	1.00	0.89	0.94	0.89

Causes of Error. To better understand RASSAR errors, we classified them into six categories (Table 3.5).

About half of all errors arose from misclassifications by the YOLOv5 object detector, in which the model

Table 3.5: Sources of errors in the RASSAR technical evaluation.

Error source	Description	Count	Share (%)
RoomPlan misclassification	RoomPlan API misclassified an object	42	32.8
RoomPlan measurement	RoomPlan API provided an inaccurate object measurement	13	10.1
RoomPlan miss	RoomPlan API missed an object	11	8.6
YOLO misclassification	YOLO misclassified an object or falsely reported a non-existent object	55	43.0
YOLO miss	YOLO missed an object of interest	3	2.3
Raycasting error	The raycasting process failed, causing an object to be missing or misplaced	4	3.1

reported an incorrect category for a detected object. Because indoor spaces contain many visually similar objects, this is a significant technical challenge. For example, condiment bottles were misclassified as medicine, drawer handles as grab bars, and bed sheets as rugs, leading to invalid results. These errors usually produced false positives, which users could address through manual checking and filtering, as discussed in Section 3.2.3.3.

The other primary cause of error was RoomPlan’s inaccurate object-category labels, which led RASSAR to overlook related accessibility issues. For example, RoomPlan classified a kitchen counter as storage, causing RASSAR to miss the corresponding counter-height issue.

3.2.5 Study 3: User Study

Finally, we performed a user study with six participants representing several stakeholder groups: two wheelchair users, one of whom was also an older adult; one new parent; one occupational therapist (OT); and one person with severe low vision who was legally blind. We also recruited one participant who did not identify with these stakeholder groups because we envisioned that people such as short-term-rental hosts might also use RASSAR; see Table 3.6. Among the wheelchair users, P1 used a manual wheelchair and P4 used a power wheelchair; both had high upper-body motor function. We recruited participants through mailing lists, outreach to local disability groups, and snowball sampling.



Figure 3.8: Study 3 participants conducting RASSAR scans.

3.2.5.1 Procedure

For the user study, two researchers visited each participant’s home for a three-part investigation. First, one researcher manually conducted a ground-truth inspection, similar to Study 2, in selected rooms (e.g., the kitchen, living room, and bathroom). Second, we provided participants with an iPhone 13 Pro Max on which RASSAR was installed. After a brief tutorial, participants independently scanned the same rooms with RASSAR while the research team observed their scanning behavior and took notes. Third, the research team conducted a semi-structured debrief interview about the overall experience and ideas for future work. The interview concluded with seven-point ratings of usability, usefulness, perceived accuracy, and willingness to use RASSAR in the future (see Table 3.6). For analysis, we compared RASSAR’s detections with the ground-truth audit, following the approach in Section 3.2.4.1, and thematically analyzed the interview data.

3.2.5.2 Findings

Overall, all six participants independently scanned their spaces with RASSAR. They gave the application mean ratings of 5.5 out of 7 for usability and 5.8 out of 7 for perceived accuracy. Below, we report technical performance, usability, usefulness, participant-suggested improvements, and potential applications.

Technical Performance. The scans achieved a mean precision of 0.79 and recall of 0.73, compared with 0.86 and 0.83, respectively, in Study 2. This difference may reflect that a single experienced member of the research team conducted the Study 2 scans, whereas participants conducted the scans in the current study. Participants also rated their perception of RASSAR’s detection performance highly ($M=5.8$ out of 7; $min=5$). P5 remarked, “The 3D reconstruction results are way better than my imagination.” Participants completed their RASSAR scans in 3.3 minutes on average, compared with 11.8 minutes for the manual

Table 3.6: Participant demographics, scan performance, and ratings in the RASSAR user study. Ratings use a seven-point scale; N/A indicates that the participant did not provide a rating.

ID	Stakeholder identity	Home type	Scan area (m ²)	Rooms scanned	Manual audit time (min)	Scan time (min)	Prec.	Rec.	Acc.	Ratings (1 to 7)			
										App usability	Usefulness of results	Detection performance	Willingness to use
P1	Wheelchair user	House	120	4	12	5.02	0.74	0.67	0.54	6	4	6	7
P2	OT	Apartment	43	2	21	3.08	0.54	0.88	0.5	5	4	5	1
P3	None	House	60	2	14	2.25	0.92	0.85	0.79	5	6	6	4
P4	Wheelchair user, older adult	Apartment	70	4	10	3.25	1	0.73	0.73	6	6	7	6
P5	New parents	House	150	3	7	3.38	0.67	0.46	0.38	6	N/A	5	5
P6	BLV	Apartment	55	3	7	3.05	0.88	0.78	0.7	5	1	6	6
Average					11.83	3.34	0.79	0.73	0.61	5.5	4.2	5.83	4.83

ground-truth audits.

Usability. As noted above, all six participants completed their home scans with RASSAR and rated the application highly usable ($M=5.5$ out of 7; $min=5$). We observed different scanning practices that may also have affected technical performance. P3, P4, and P6 repeatedly tilted the phone up and down during their scans, whereas P2 and P5 held the phone vertically and therefore missed accessibility and safety issues on the floor. Visual or audio cues that identify unscanned surfaces could help prevent this oversight. P6, the participant with low vision, completed the scan using the audio-assistance feature. Other participants also found the audio assistance useful. P4 and P5, both of whom were sighted, enabled audio feedback during their RASSAR scans and felt that it increased their awareness of scan progress and detected issues. P5 explained, “The audio helped me a lot in understanding what’s happening and what should I focus on during the scan.”

Usefulness. Ratings of usefulness were more nuanced. P3 and P4 each rated usefulness as 6 out of 7, explaining, “I would never know that the throw rug could be an issue” (P3) and “These results will be potentially useful if I want to pick a new home” (P4). P1 and P2 each rated usefulness as 4 out of 7. P1, a manual-wheelchair user, felt that the ADA-based issues did not match her needs: “My house is very usable to me, but in RASSAR’s head, it wasn’t.” Nevertheless, P1 rated the application highly usable and wanted to use it in the future. P2, the OT, wanted RASSAR to be more customizable, stating that “it’s not customized enough.” P6, who had low vision, gave high ratings in all categories except usefulness. He found the detected issues interesting and helpful but rated usefulness low because the final scan-summary interface was not sufficiently accessible to him.

Potential Applications. In the debrief interviews, participants offered a variety of use cases for RAS-

SAR. P1, who gave the highest rating for willingness to use RASSAR, envisioned it as a way to scan, assess, and share the accessibility of public spaces: “It would be really helpful in public spaces. I’m on several Facebook groups for people with disability. And we will all turn to write reviews of different places that we go. [With RASSAR], we can have a more objective way to share that information.” Similarly, P2, the OT, felt that RASSAR could raise awareness of accessibility issues and reach more people than on-site OT inspections: “What OT can do [to help people] is very limited. There are so many people in need but very few have access to OT service.” Although P2 gave a low rating for her own willingness to use RASSAR, she was willing to recommend it to others: “I might not use it since I already have this knowledge. But I would recommend it to my family or anyone in need. It’s so convenient and way better than me lecturing them!”

P3 identified more personal use cases: “If I move to a new home, or when I get injured someday, I might want to use this (to check for home safety).” Similarly, P4 thought that RASSAR could help people audit prospective homes before making a purchase: “It would be helpful for auditing another place.” P5 thought that RASSAR could support childproofing if future versions included more issues related to children.

System Improvements. Participants suggested several ways to improve interaction with the system. Suggestions for the summary view (Figure 3.6) included making the 3D model easier to manipulate (P1), organizing scan results by room (P2), and offering a list view (P2 and P5). P6 also requested a list view because it would be more compatible with screen readers. Other recommendations concerned scanning support. P5 suggested audio or visual cues that would alert users when they missed an indoor surface, such as the floor. P6 requested proximity alerts to help people who are blind or have low vision avoid colliding with barriers while scanning. P3 advocated for more direct visual guidance, such as arrows, to help users conduct thorough scans without overlooking key areas.

3.2.6 Discussion

We introduced the first mobile AR system to detect accessibility and safety issues in residential spaces. Using RASSAR, individuals can semi-automatically audit their homes and generate real-time 3D reconstructions that highlight potential issues. We also conducted three studies to examine real-world needs, technical performance, and user experience. Below, we discuss the implications of our findings and opportunities for future work.

3.2.6.1 Application Scenarios

Unlike prior work in automatic accessibility auditing, which often involves complex data-collection processes [Balado et al., 2017; Fu et al., 2020] or specialized hardware [Serna and Marcotegui, 2013; Ayala-Alfaro et al., 2021], RASSAR is a smartphone application. Its availability on a mobile device broadens its potential applications. Through the formative study (Section 3.2.2.2) and user study (Section 3.2.5), we identified several possible use cases, which we discuss below.

Prior-Visit Auditing. One common challenge identified in the formative study was uncertainty about a location’s accessibility before a visit. Participants raised this concern for rental accommodations, such as hotel rooms and short-term rentals; public spaces, such as restaurants and shops; and the homes of friends or family members. Based on our user-study feedback (Section 3.2.5), RASSAR could support a standardized scanning and detection process for previewing potential accessibility and safety issues. Site owners could use RASSAR to assess whether their spaces meet general accessibility requirements and share scan results so that prospective visitors can evaluate a space for themselves. Future work should examine interfaces for sharing and interpreting these results.

Improving Spaces to Accommodate Life Changes. Changes such as illness, childbirth, or caring for an older family member can alter a household’s accessibility and safety needs. Our studies suggest that RASSAR could help residents identify potential risks during these transitions and consider changes such as removing hazardous items or adjusting object dimensions.

Complementing Occupational Therapists’ Home Visits. As prior research [RENDAL and LAPE, 2018] and our studies with occupational therapists indicate, requesting, scheduling, and paying for a home visit can be difficult. RASSAR could support remote assistance by allowing occupational therapists to review scan results that include visual information, 3D positions, and measurements. Compared with a video call alone, this representation could make remote auditing more structured and provide additional spatial evidence for review.

Beyond Residential Spaces. Although RASSAR is designed for residences, its technical framework could be applied to spaces such as offices, schools, and restaurants. We conducted initial tests in two offices. Future work could incorporate rubrics for nonresidential spaces and examine methods for uploading, sharing, and viewing assessments.

3.2.6.2 Detection Performance

As described in subsubsection 3.2.4.3 and subsection 3.2.5, RASSAR achieved a mean precision of 0.86 and recall of 0.83 when operated by our research team, compared with 0.79 and 0.73, respectively, when operated by study participants. Participants completed RASSAR audits in 3.3 minutes on average, compared with 11.8 minutes for manual audits. Although preliminary, these results are promising. RASSAR’s performance could still be improved. Most errors arose from limitations in our YOLOv5 model (Section 3.2.3.1) and the RoomPlan API. Future work could expand the object-detection model to cover furniture categories and thereby correct RoomPlan misclassifications, as well as expand the training dataset for small indoor objects. Further study is needed to determine the accuracy required for each application scenario proposed above.

3.2.6.3 Accessibility Issue Scope

RASSAR currently detects 20 types of accessibility and safety issues across four categories (Figure 3.4). Future versions could expand this list based on needs identified in the literature (e.g., detecting sharp furniture edges [Niemeyer and Vogel, 1994]), the formative study (e.g., home entrances, stairs, and bathing facilities), and the user study (e.g., wheelchair maneuvering spaces).

3.2.6.4 Potential Beyond the ADA

The current RASSAR system relies primarily on ADA design guidelines as a source for its accessibility rubrics. However, as both our formative and summative studies showed, these guidelines specify minimum requirements and may not fit every individual’s needs. P2 explained, “ADA is just a fixed standard for reference, thus it cannot ensure fitting on everyone.” Similarly, a wheelchair user in Study 3 recalled that a hotel room not designated as ADA accessible worked well for her when the designated rooms were unavailable. In this case, a binary classification of “ADA accessible” excluded a space that met her individual needs. RASSAR’s custom JSON-based rubric definitions could support assessments personalized to those needs. Future work should explore interfaces for authoring personalized rubrics.

3.2.7 Conclusion

We introduced RASSAR, a mobile AR system that semi-automatically audits residential spaces for accessibility and safety issues. RASSAR uses mobile LiDAR and computer-vision models to identify, localize, and display potential issues and provide recommendations for mitigation. Users can customize the audit by specifying target accessibility communities and can verify or remove detected issues. Our technical evaluation (Study 2) in ten home spaces characterized RASSAR’s detection performance, consistency, and auditing time. Our initial user study (Study 3) further demonstrated RASSAR’s potential among stakeholder groups. This work advances indoor accessibility and safety auditing, contributes an object-detection model and dataset to the research community, and opens new research directions for human-AI collaborative indoor auditing.

3.3 Frontier Synthesis

RASSAR shows that real-time spatial assistance is not a single prediction delivered to a user. It is a loop among guided sensing, machine interpretation, human verification, and persistent representation. The formative study demonstrates why this loop must be personalized: participants’ practical accessibility knowledge often exceeds or differs from generic standards. The technical evaluation demonstrates that a mobile system can detect many relevant issues quickly, while its error analysis shows why automated findings cannot be treated as complete. The user study further shows that usability and usefulness are distinct. Participants could operate the system successfully while still questioning whether a general rubric reflected their lived needs or whether the review interface was accessible to them.

This frontier advances human-AI symbiotic spatial perception in two ways. First, it enlarges a person’s perceptual input at the moment of action by making otherwise easy-to-miss dimensions, objects, and risks visible. Second, it identifies the conditions under which this enlargement remains accountable: users must be able to see where a finding is anchored, understand why it was produced, correct it, and adapt the criteria being applied.

Chapter 4

Frontier II: Spatial Perception Beyond Physical Presence

4.1 Preamble

The second frontier concerns the *reach* of spatial perception. The thesis so far has considered assistance at the point of action; many important decisions, however, must be made before a person can arrive at a place or after the moment of observation has passed. A traveler may need to assess a room before booking it, a facilities manager may need to inspect equipment across campus, and a traveler may need to understand a complex transportation facility which they never visited in order to catch their train/flight. Distance, time, mobility, danger, privacy, and the cost of repeated visits all constrain direct observation. Extending perception beyond physical presence therefore means collecting spatial evidence on a person's behalf and preserving it for later or remote judgment.

This goal is more demanding than remote viewing. A photograph captures appearance from one viewpoint, video records a path through a space, a floor plan abstracts connectivity, and a 3D reconstruction permits free-viewpoint exploration. Each is useful, but none inherently preserves the objects, dimensions, routes, and functional relationships needed for every decision. Remote accessibility systems demonstrate both the value of mediated inspection and the need for task-appropriate representations: people can explore virtual replicas, request personalized scans, or preview routes through street-level imagery, while still need-

ing ways to recognize missing or questionable evidence [Pei et al., 2023; Huang et al., 2025; Jain et al., 2026].

Automation can broaden this spatial and temporal reach. Robots can repeat structured scans, drones can observe tall or expansive interiors, and reconstruction methods can turn image collections or video into persistent geometry [Wang et al., 2025a; Xie et al., 2025]. Yet remote capture redistributes rather than removes human work. The sensing platform must move safely and cover relevant viewpoints; reconstruction must tolerate occlusion, motion, changing illumination, and scale; and the resulting map must remain meaningful to people who did not witness its capture.

The central challenge is relevance. A fully autonomous system can collect extensive geometry while missing critical facilities, constraints, or semantics that can help determine a spatial decision; exhaustive manual annotation, in contrast, defeats the purpose of automation. This frontier investigates how sensing systems and lightweight human input can complement one another in building inspectable, task-relevant spatial representations. The two projects in this chapter examine complementary points in that design space.

4.2 RAIS: Robotic Accessibility Indoor Scanning

RAIS investigates whether the evidence-collection burden in RASSAR can be transferred from a person to a robot. It integrates RASSAR with a commodity robotic vacuum, phone, gimbal, and supporting sensors. The prototype follows walls, constructs a 2D occupancy map, collects a phone-based room reconstruction, and evaluates the resulting model for accessibility concerns. Because RASSAR’s rubric and reporting components already establish how room geometry becomes an accessibility assessment, this section focuses on the new problem introduced by the frontier: autonomous movement and capture.

4.2.1 Introduction

Despite significant recent progress [Cai et al., 2024; Pei et al., 2023; Su et al., 2024d; Yoon et al., 2019; Simões et al., 2020; Su et al., 2024b], mapping and assessing indoor spaces and creating personalized routes through them remain grand challenges in accessibility research [Froehlich et al., 2019; Striegl et al., 2020]. Advances in photogrammetry, camera hardware, and laser scanning have enabled companies such as Matterport [Inc., 2024c], Metareal [Inc., 2024d], and EyeSpy360 [Inc., 2024a] to create precise, immersive

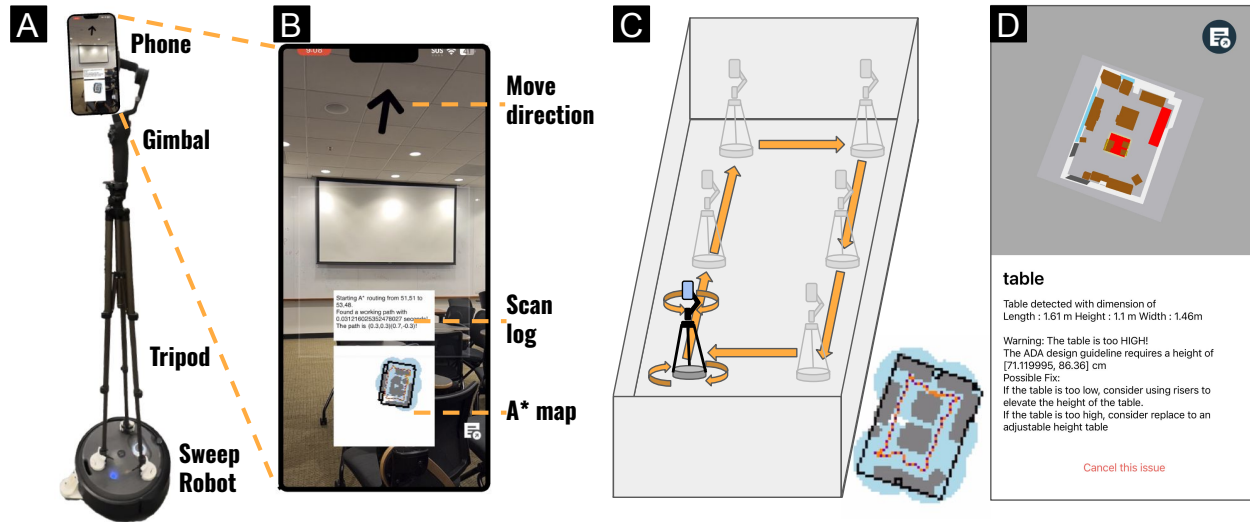


Figure 4.1: RAIS combines commodity robotic hardware with mobile sensing to automate indoor mapping and accessibility assessment. The system produces a navigation map, captures a parametric room model, and marks detected accessibility issues.

3D reconstructions of indoor spaces for virtual tours and other mapping applications. Other systems use wheeled carts equipped with scanning sensors, such as FARO’s Focus Swift [Aniwa, 2020], or large wearable backpack systems, such as the NavVis VLX 3 [NavVis, 2024]. These systems offer high-fidelity scans for blueprint generation, construction tracking, and virtual walkthroughs, but they do not focus on identifying and evaluating accessibility features such as narrow hallways, staircases, and missing assistive aids. The hardware is also costly, and scans can be time-consuming, expensive, and dependent on professional training. Scaling these systems to identify and track accessibility problems throughout large buildings, such as offices, airports, and train stations, can therefore be cost-prohibitive.

Drawing on recent work in robotics for automated indoor scanning and inspection [Robotics, 2024; Inc., 2024b] and emerging work in smartphone-based indoor accessibility mapping [Yoon et al., 2019; Su et al., 2023, 2024d], we introduce RAIS (Robotic Accessibility Indoor Scanner), a robotic indoor-mapping and accessibility-assessment system (Figure 4.1). This rapid prototype uses off-the-shelf components, including a vacuum robot, smartphone, and phone gimbal, and runs RASSAR, an application for 3D reconstruction and indoor accessibility auditing [Su et al., 2024d]. In our initial tests, RAIS independently navigated previously unmapped indoor spaces and detected potential accessibility issues in real time. It generated parametric 3D map reconstructions that supported both in situ and post hoc accessibility audits.

To evaluate RAIS, we performed a pilot study in three indoor environments: a makerspace, a classroom, and an office. In each environment, we placed RAIS at a random position, started a scanning session, and allowed it to conduct the reconstruction and accessibility scan independently. RAIS completed an automated scan in all three environments, generating 3D reconstructions that included architectural features (walls, doors, and windows) and furniture and highlighting potential accessibility issues.

In summary, this project contributes a low-cost prototype for robotic mapping and indoor accessibility assessment built from commodity hardware. Our overarching goal is to develop automated indoor-mapping techniques that prioritize accessibility and support remote visit planning, in situ navigation along accessible routes, and infrastructure maintenance.

4.2.2 The RAIS Prototype

To examine the feasibility of a robotic indoor-mapping and accessibility-scanning system, we rapidly constructed a hardware-software prototype using off-the-shelf components. We first describe the key design components and then explain the RAIS mapping and scanning process.

4.2.2.1 Design and Components

As a rapid prototype, RAIS is built with commodity hardware and software components:

LiDAR-equipped smartphone: To scan and reconstruct 3D environments, we use an iPhone 13 Pro Max. The map reconstructed in real time also guides the robot’s path.

Robot: To autonomously navigate and map indoor environments, we use an iRobot Create 3 wheeled vacuum robot.

Motorized phone gimbal: To control the smartphone’s capture angles and ensure broad coverage, the phone is mounted on a DJI Osmo Mobile 6 gimbal that rotates left, right, up, and down during scanning.

Tripod: We elevate the gimbal and phone with a tripod to improve capture of indoor objects.

Software: Our previously published RASSAR system [Su et al., 2023, 2024d] performs 3D reconstruction and scans for accessibility barriers in real time. RASSAR is a custom iPhone application that uses LiDAR and a custom-trained computer-vision model to identify, localize, and visualize indoor accessibility and safety issues, such as inaccessible table heights and loose rugs. We also developed iPhone software to

route the RAIS robot and control the gimbal, as described below.

4.2.2.2 Scanning Process

To enable RAIS to explore and scan unknown indoor spaces, we implemented SLAM and a routing strategy. We use Apple’s ARKit and the RoomPlan API [Apple, 2022] to create a real-time 3D reconstruction of indoor spaces and locate the robot’s 3D coordinates. The exploration and navigation strategy combines a wall-following method that guides the robot into unknown spaces with an A* pathfinding algorithm that avoids obstacles.

Algorithm 1 RAIS wall-following strategy.

```

1: while not back at initial position do
2:   if farther than 2.5 m from all scanned positions then
3:     Rotate robot and gimbal to perform scan at current position
4:   end if
5:   if not following any wall then
6:     Find nearest wall and start following it
7:   end if
8:   Move along the following wall
9:   if new wall detected within 1.2 m then
10:    Follow the new wall
11:   else if left-side wall ends then
12:    Turn left and move forward until a new wall is found
13:   end if
14: end while
15: end algorithm

```

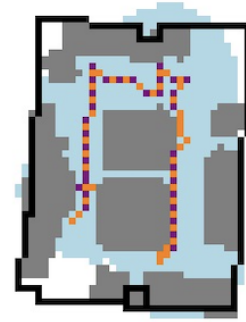


Figure 4.2: Example of RAIS’s A* routing. Grid cells measure 20 cm × 20 cm. Black represents walls, gray represents furniture, blue represents scanned space, orange represents the path, and purple represents stopping positions.

For exploration, we use a classic wall-following algorithm [van Turenout et al., 1992]. The robot begins away from a wall while the phone performs an initial scan to create a partial map. RAIS selects the nearest wall and follows it at a distance of 1.2 meters. When it detects a new wall within 1.2 meters, RAIS follows that wall instead. This process continues until the robot returns to its initial position and closes its trajectory loop, at which point the scan is complete (Figure 1).

Because furniture and clutter may obstruct the robot’s path, we use the well-established A* pathfinding algorithm to avoid obstacles [ElHalawany et al., 2013]. We rasterize the real-time 3D reconstruction from

RoomPlan to create a grid map. The robot’s current location and moving goal are projected onto this map for A* search, and the resulting path is executed by the robot. An example is shown in Figure 4.2.

4.2.2.3 Scanning Results

After a scan, RAIS represents the room layout and indoor objects, such as tables and chairs, as a parametric 3D model. The model includes architectural features such as walls, doors, and windows, as well as furniture. An example reconstruction appears in Figure 4.1D. RAIS reuses the RASSAR visualization interface [Su et al., 2024d], and users can download the model for inspection in other computer-aided design programs. Accessibility rubrics allow RAIS to identify potential issues either during scanning or through post hoc analysis.

For this initial prototype, we adopt four accessibility rubrics from the ADA design guidelines [of Justice, 2024]: (1) door width, requiring more than 32 in. (81.3 cm) of clearance for wheelchair passage; (2) table or counter height, requiring a height between 28 and 34 in. (71.1 to 86.4 cm); (3) cabinet height, requiring a reachable height below 27 in. (68.6 cm); and (4) routing space, requiring more than 36 in. (91.4 cm) of width for wheelchair passage. The first three issues can be detected automatically in real time during scanning. Routing space is currently reviewed manually and will be automated in future work.

4.2.3 Preliminary Evaluation

To evaluate RAIS’s performance, we conducted test scans in three indoor environments: a makerspace, a classroom, and an office (Table 4.1). In each space, we deployed RAIS at a random location and started an automated scan. The robot then maneuvered through the space, generated a 3D reconstruction and 2D map, and conducted an accessibility scan. We timed and recorded the entire process. For comparison, we manually inspected each environment to establish ground truth. Table 4.1 reports the space characteristics, scan times, and accessibility-issue detections. Figure 4.3 shows the environments, generated maps, and scan routes.

RAIS completed scans in all three spaces, taking 8.5 minutes on average. It detected six of the eight applicable accessibility issues (75%), including inaccessible table and counter heights in the makerspace and office and narrow corridors in the classroom and office. RAIS did not identify an inaccessible cabinet

Table 4.1: RAIS test spaces and results. Accessibility issues are identified by their index in Section 4.2.2.3.

Space	Area (m ²)	Scan time (min)	Time per m ² (s/m ²)	Issues in space	Issues detected
S1: Makerspace	40	10	15	i2, i2, i2, i3	i2, i2
S2: Office	32	6.5	9.75	i2, i2, i4	i2, i2, i4
S3: Classroom	48	9	11.25	i4	i4

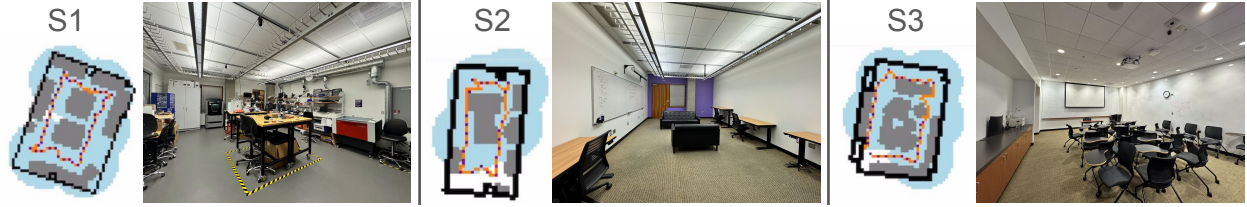


Figure 4.3: Preliminary RAIS evaluation in three indoor spaces: a makerspace (S1), an office (S2), and a classroom (S3). For each space, the figure shows a photograph, the automatically generated 2D map, and RAIS’s scanning route.

in the makerspace. Although preliminary, these results establish the feasibility of the prototype in the three test environments.

4.2.4 Discussion and Future Work

RAIS explores one approach to collecting data for indoor accessibility maps. Its parametric 3D reconstructions could inform infrastructure improvements and provide spatial information for people with disabilities. We built RAIS to examine the feasibility of automatically creating, updating, and maintaining indoor maps focused on accessibility. The preliminary evaluation demonstrated the prototype’s potential and represents an initial step toward addressing the broader challenge of indoor accessibility mapping.

Because RAIS operates autonomously and can explore previously unmapped spaces, future versions could support repeated scanning in changing environments, such as malls and public transportation stations. Regular scans could help keep indoor accessibility maps current and could also provide building managers with information relevant to cleanliness and safety. These applications require evaluation at a scale and level of environmental complexity beyond the current pilot study.

As a proof-of-concept prototype, RAIS demonstrates autonomous scanning capabilities but leaves several research questions unanswered. Can its sensing components be integrated with widely used indoor

robots, such as floor scrubbers? Can RAIS robustly and efficiently scan larger spaces, such as malls and train stations? How should scan results be inspected, tracked, archived, and shared to improve understanding of indoor accessibility while supporting facility maintenance? Future implementations and studies should address these questions.

4.2.5 Conclusion

In this project, we presented RAIS, an initial step toward automated indoor accessibility mapping. RAIS scans previously unmapped indoor environments, creates parametric 3D reconstructions and 2D maps, and identifies potential accessibility issues using LiDAR and computer vision. Our preliminary evaluation in three indoor spaces established the feasibility of this commodity-hardware prototype. The project points toward systems that could conduct repeated scans with limited manual control and maintain up-to-date accessibility maps for indoor navigation and facility maintenance.

4.3 From Room-Scale Auditing to Large-Space Semantic Mapping

RAIS demonstrates that autonomous capture can reduce the need for a person to conduct a room scan, but it also exposes the limits of a fixed, room-scale collection strategy. Wall following and occupancy mapping are well suited to bounded rooms and accessibility criteria that can be computed from a parametric model. Larger public and institutional spaces introduce open atria, multiple floors, elevated facilities, irregular trajectories, and a wider range of stakeholder questions. Their useful representation must contain not only geometry but also semantic points of interest such as doors, stairs, elevators, ramps, alarms, and equipment.

FlyMeThrough addresses this broader setting. Rather than assume that the system can infer every meaningful object autonomously, it asks occupants and building managers to provide sparse clicks and labels in familiar video frames. AI propagates those annotations and converts them into 3D object anchors. The connection between the projects is therefore a progression in both scale and human-AI division of labor: RAIS automates a predefined room audit, while FlyMeThrough combines flexible capture with human guidance about what the map should remember.

4.4 FlyMeThrough: Human-AI Collaborative Indoor Mapping

FlyMeThrough uses commodity-drone video, structure-from-motion reconstruction, video segmentation, depth-guided ray casting, and interactive review to produce semantic 3D maps of large indoor spaces. Its object-centered annotation workflow treats a click as compact spatial evidence: the user identifies a meaningful facility in one frame, SAM2 propagates the mask through nearby frames, and the system projects the observations into the reconstructed world. The resulting bounding box becomes a persistent point of interest that other users can inspect from a free viewpoint.

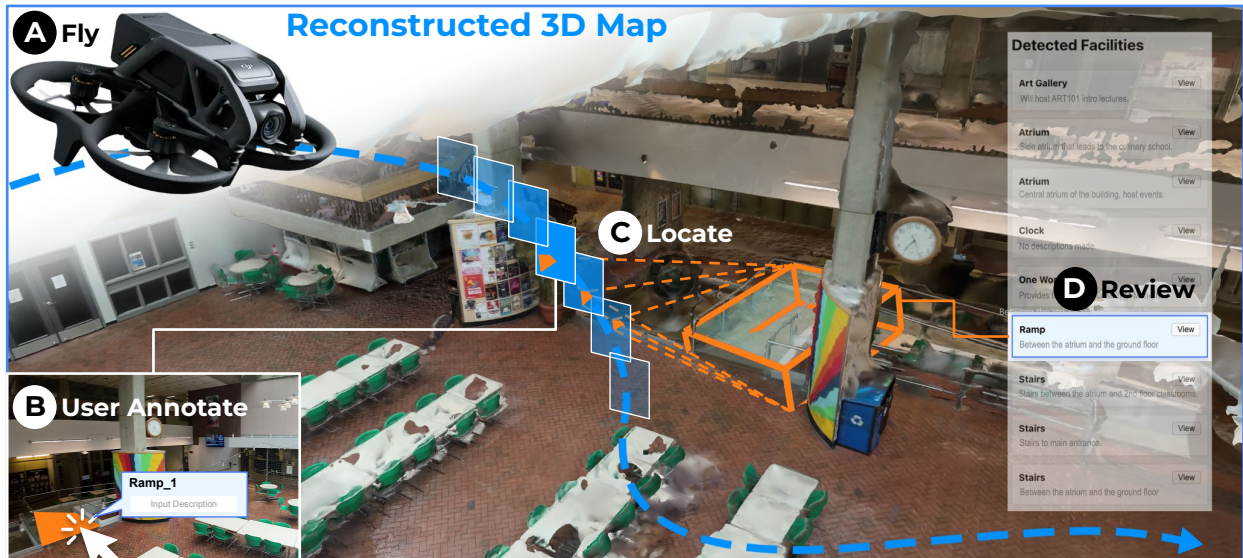


Figure 4.4: FlyMeThrough reconstructs large indoor spaces from commodity-drone video. Sparse user annotations are propagated across frames, localized in the 3D model, and exposed as interactive facility anchors.

4.4.1 Introduction

Indoor mapping data for public spaces, such as office buildings, train stations, airports, and stadiums, is crucial for navigation, route planning, space evaluation, and tracking spatial changes over time. However, such data is often scarce and outdated [Pipelidis et al., 2019]. Mainstream indoor-mapping procedures either transform CAD building plans into digital maps or deploy dedicated scanning hardware and services. Both approaches require high data-acquisition costs and long timelines [Yue and Wu, 2018], making them difficult to deploy and maintain at scale.

In contrast, drone-based indoor mapping uses drones to capture key spatial information throughout large indoor spaces and can reduce data-collection and maintenance effort. Academic research [Karam et al., 2022; Zhou et al., 2014] and commercial products such as Skydio [Skydio, 2025] and Elios [Flyability, 2025] demonstrate its feasibility. However, existing solutions often rely on LiDAR-equipped drones that are prohibitively expensive for large-scale deployment; for example, an Elios 3 LiDAR drone costs \$50,000. Moreover, the resulting maps lack points of interest (POIs) that building managers and visitors need. To address these challenges, we develop a more affordable, scalable mapping pipeline that uses only RGB video and can therefore operate with commodity drones. We also introduce a human-AI collaborative annotation and localization process that lets building-management staff identify and mark key indoor POIs, such as entrances, stairs, and elevators, making the resulting 3D maps more useful for navigation, routing, and space evaluation.

We present FlyMeThrough, a human-AI collaborative indoor-mapping system with an end-to-end pipeline that transforms RGB footage from indoor drone flights into POI-enriched 3D maps. FlyMeThrough has three major technical components: structure-from-motion (SfM) reconstruction, which estimates video-frame camera positions and constructs 3D meshes; a human-AI collaborative annotation pipeline, which lets users annotate and localize key indoor POIs; and a web interface for reviewing the final 3D models and POI bounding boxes.

To evaluate FlyMeThrough, we collected drone footage from 12 indoor spaces of varying sizes and functions with the approval and guidance of the building managers responsible for those spaces. We also conducted a user study with five of these building managers and five active building occupants to assess the annotation interface. Participants reviewed the drone-based mapping process, annotated key indoor POIs in the collected data, and discussed the interface’s usability and potential applications. The results demonstrate the technical feasibility of the mapping pipeline and generally positive perceptions of its usability while identifying opportunities for improvement.

Our contributions are threefold. First, we implement a drone-based indoor-mapping system that uses only RGB data and can therefore employ affordable commodity drones. Second, we introduce a human-AI collaborative pipeline for creating POIs in reconstructed indoor 3D maps. Third, we report insights from a user study with building managers and occupants that can guide future drone-based indoor mapping.

4.4.2 The 3D Mapping of Indoor Spaces

FlyMeThrough is a multi-stage system that transforms RGB drone footage into interactive 3D indoor maps annotated with key points of interest (POIs) (Figure 4.4). The system begins with drone-based RGB video collection, followed by structure from motion (SfM) to estimate camera trajectories and reconstruct the space as a 3D mesh. Users such as building managers annotate key facilities, such as stairs, doors, and elevators, on selected video frames using their grounded knowledge of the space. SAM2 [Ravi et al., 2024a] expands these annotations into per-frame segmentation masks, and a depth-guided raycasting algorithm projects the masks into 3D space. The result is a semantically enriched 3D map with interactive POI bounding boxes that indicate the location and dimensions of each annotated facility.

This work extends an earlier poster prototype that explored the feasibility of using commodity drones for indoor 3D mapping and automatic detection of facilities such as stairs, doors, elevators, and ramps. While retaining RGB video capture and SfM-based reconstruction, FlyMeThrough introduces a more robust human-AI collaborative architecture. Its annotation pipeline and depth-based localization improve the semantic fidelity of the 3D maps, enabling end users to embed local spatial knowledge in the resulting reconstruction.

4.4.2.1 Drone-Based Data Collection

In line with prior work, we use RGB-only video input for its practicality and compatibility with consumer drone hardware. For safety and controllability during indoor flights, we use a DJI Avata drone, purchased for \$759 in 2024, because of its propeller guards and immersive controller headset. We select its wide-angle camera setting, which yields a 110-degree field of view, and capture indoor-flight video at 4K resolution and 30 frames per second. The lead author manually controls the drone to fly steadily through the public open spaces of the tested buildings.

4.4.2.2 SfM Indoor Reconstruction

Following the settings from our previous work, we extract frames from drone footage at 2 FPS for indoor reconstruction. We compared state-of-the-art open-source approaches, including NeRF [Mildenhall et al., 2021; Tancik et al., 2023], Gaussian Splatting [Kerbl et al., 2023], and learning-based methods such as MV-

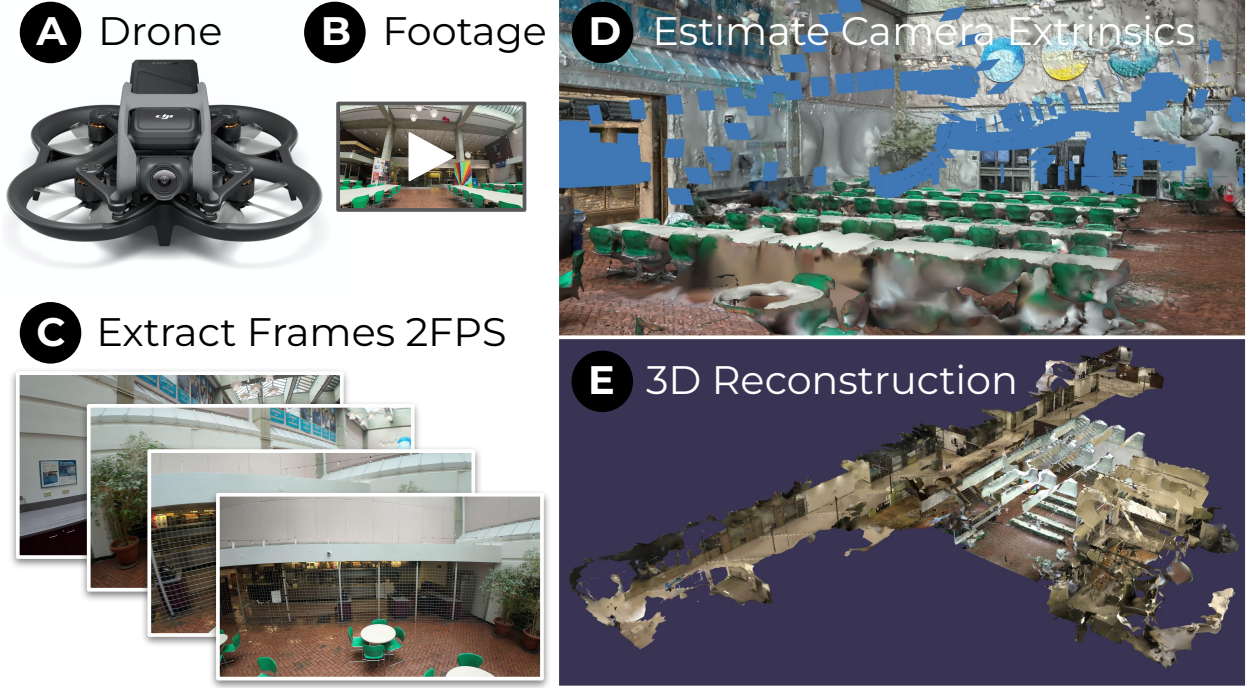


Figure 4.5: Drone-based indoor scanning and reconstruction. (A) The DJI Avata drone. (B) We collect flight footage as 4K, 30-FPS video. (C) We extract frames at 2 FPS. (D) Agisoft Metashape estimates the camera extrinsics of extracted frames and (E) reconstructs the space in 3D.

DUST3R+ [Tang et al., 2024] and MAST3R-SLAM [Murai et al., 2024], then selected the option that best balanced output quality, success rate, and computing requirements. We use the photogrammetry software Agisoft Metashape [Over et al., 2021] for SfM camera-pose estimation and 3D reconstruction. Figure 4.5E shows the reconstructed 3D model, while Figure 4.5D shows estimated camera poses and the flight trajectory.

4.4.2.3 Indoor Facility Segmentation

To embed key facility information in the reconstructed 3D map, FlyMeThrough employs the SAM2 [Ravi et al., 2024a] video-segmentation model to interactively identify and label indoor objects. For each scene, users click video frames to annotate objects, including predefined types drawn from existing indoor-scanning work [Su et al., 2024d], such as doors, elevators, ramps, and stairs, as well as custom categories that users define.

This design marks a departure from our previous system (poster omitted for anonymity), which relied

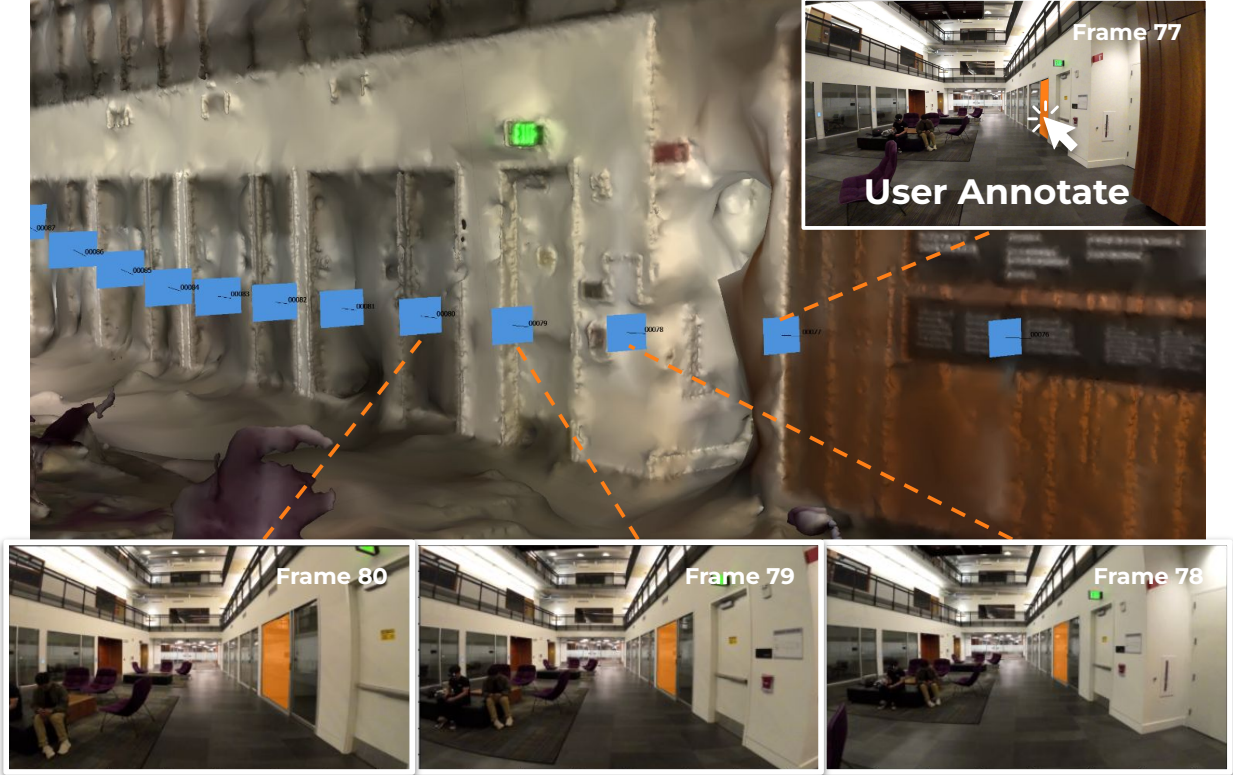


Figure 4.6: SAM2 segments key indoor facilities. When a user annotates one frame, the mask is propagated to subsequent frames.

on YOLOv8 [Ultralytics, 2023] for fully automated POI detection. While effective in automating object recognition, YOLOv8 lacked flexibility for open-vocabulary object types and precision for multi-frame consistency, especially in novel or complex indoor scenes. In contrast, SAM2 offers several key advantages that informed our decision to adopt it:

Interactive and Efficient Annotation: SAM2 natively supports interactive prompts, such as point clicks and bounding boxes, that integrate with the UI and minimize annotation effort. With minimal user input, the model generates high-quality pixel-level masks, significantly reducing manual effort.

Generalization and Adaptability: Unlike YOLOv8’s closed-set object categories, SAM2 generalizes across diverse object types and indoor layouts. Its zero-shot segmentation capability supports accurate labeling of novel instances, making it suitable for dynamic, unstructured indoor settings and diverse user needs.

Precision and Continuity: SAM2 produces pixel-accurate segmentations that capture fine object bound-

aries. This granularity is critical for downstream ray-casting-based localization, where geometric precision determines placement accuracy in 3D space. Propagating segmentations across adjacent frames also improves temporal consistency for the subsequent localization step.

In deployment, we use a pretrained SAM2 checkpoint [AI, 2024], which tokenizes input video frames and infers object masks in subsequent frames from a user segmentation (Figure 4.6). We enhance segmentation consistency with a local temporal-control strategy: predictions from a user annotation are retained only for consecutive frames and terminate when the target object exits the view. This helps mitigate missegmentations caused by repetitive layouts, a common challenge in hallways and mirrored environments.

4.4.2.4 Localizing Indoor Features with Depth-Guided Ray Casting

The segmentation results, as masks over sequential video frames, are projected into the reconstructed 3D maps for localization. In our previous system, this projection was performed by individual pixels following the classical pinhole camera model [Sturm, 2021]. Each masked pixel was back-projected through the camera’s intrinsic and extrinsic parameters to cast a ray into the 3D scene. Intersections between these rays and the reconstructed surface mesh were computed, and the resulting hit points were post-processed via spatial clustering to generate 3D bounding boxes for each object instance. However, this approach was highly sensitive to mesh artifacts, such as surface holes and noise, which frequently led to fragmented or inaccurate localization results.

To address this problem, we introduce a **depth-guided ray-casting strategy** that uses monocular depth estimates to constrain ray lengths within an image mask. The method creates a segmentation point cloud and casts it as a unified structure, improving localization accuracy and suppressing false intersections caused by mesh artifacts.

Classical Pinhole Projection Model. We retain the pinhole camera model, in which a 3D point $\vec{P}_w = (X_w, Y_w, Z_w)^T$ is projected to an image coordinate (u, v) using the camera intrinsic matrix K and extrinsic parameters R and t :

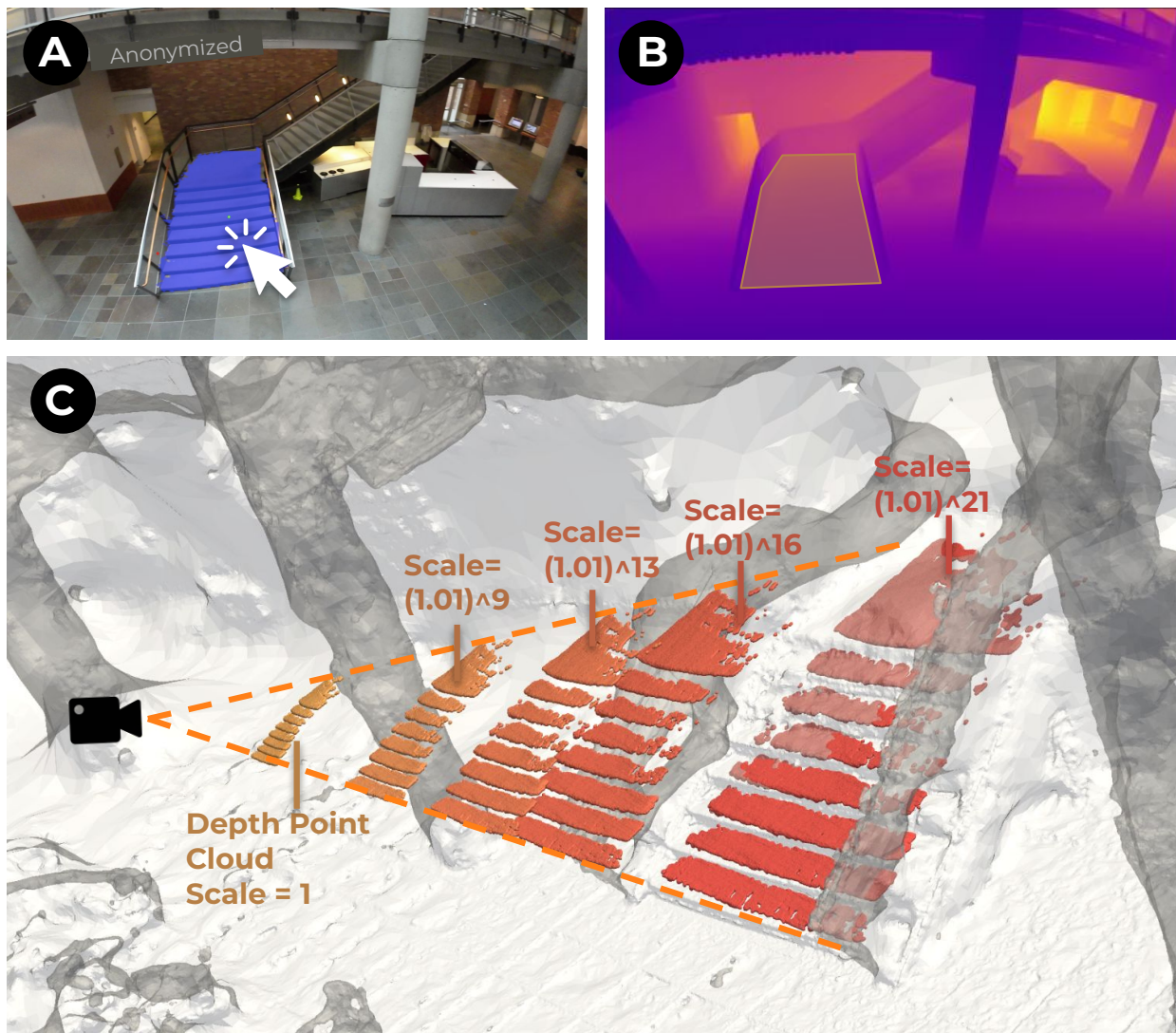


Figure 4.7: Depth-guided ray casting localizes user annotations. (A) Each annotation mask is combined with (B) an inferred depth map to produce (C) a 3D point cloud, which is iteratively enlarged until it intersects the reconstructed model.

$$Z_c \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot [R | t] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

Given known K , R , and t from the SfM reconstruction, the only missing component is the depth Z_c value, which is traditionally calculated by raycasting.

Creating a Segmentation Point Cloud with Depth Estimation. Although depth is not directly available in monocular RGB video, we estimate relative depth for each pixel in an image frame using Depth Pro, a state-of-the-art monocular depth-prediction model [Bochkovskii et al., 2024]. We then use the predicted depth values, denoted by $\hat{Z}_c$, to extrude the pixels in the segmentation mask to 3D coordinates $\tilde{P}_w$:

$$\tilde{P}_w = R^{-1} \cdot \left(\hat{Z}_c \cdot K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - t \right) \quad (2)$$

This process produces a 3D point cloud from the pixels in a 2D segmentation mask, capturing the object’s spatial extent and relative depth within the scene. Figure 4.7 shows an example. The resulting point cloud is not metrically aligned with the mesh because the monocular depth prediction has an unknown global scale factor. The following casting steps iteratively approximate that scale.

Unified Segmentation Point-Cloud Casting. Rather than casting each pixel independently, as in our prior approach, we cast the segmentation point cloud as a whole. Independent rays may stop at irregular geometry or pass through holes. In the unified approach, we multiply the point cloud’s scale by 1.01 at each iteration and compute the proportion that intersects the reconstructed mesh. Casting stops when this proportion reaches 22%, a threshold selected empirically (Figure 4.7). This strategy reduces sensitivity to local mesh defects, including holes, clutter, and irregular geometry, while preserving the object’s shape and orientation. It prevents premature or delayed intersections and mitigates offsets introduced by surface irregularities. Compared with our previous system, the casting process achieves greater geometric accuracy and semantic consistency in complex indoor 3D models.

Adaptive Downsampling for Efficiency.

To improve runtime performance during point-cloud ray casting, we introduce adaptive spatial downsampling. The number of retained points is selected dynamically to reduce point-cloud size while preserving overall shape. Larger objects retain more points to maintain precision, whereas smaller objects receive more compact representations. This balance reduces computation while preserving sufficient spatial fidelity.

4.4.2.5 Bounding-Box Creation

After raycasting the downsampled segmentation cloud into the mesh, the resulting intersection points are clustered to produce a final 3D bounding box. This bounding box captures not only the object’s location, but also its approximate size and orientation in 3D space.

Because we cast segmentation results from multiple frames of the same object to reduce noise, the combined result is a dense, overlapping point cloud. We apply DBSCAN [Ester et al., 1996] to segment and filter the cloud, removing outliers and preserving the most coherent geometric regions associated with the object. We then compute a PCA-based approximation of the minimum-volume bounding box [Dimitrov et al., 2006] for each cluster. This process produces tighter, orientation-aware volumes that better reflect the object’s geometric extent, particularly when it is not axis aligned.

4.4.2.6 Modularity of Implementation

FlyMeThrough is a multistage pipeline that combines commodity hardware, commercial software, classical algorithms, learned models, and our original methods. Each component introduces limitations and opportunities for improvement. We therefore designed the system as a modular architecture in which individual stages can be replaced as better-performing alternatives become available. The same modularity lets future users adapt the implementation to their hardware, software, and computational constraints.

4.4.3 The FlyMeThrough Interface

As described above, FlyMeThrough relies on people to contribute grounded knowledge about which indoor facilities matter and where they are located. This mapping task is difficult to automate fully for two reasons. First, indoor facilities vary in appearance, making them difficult for general-purpose models to detect

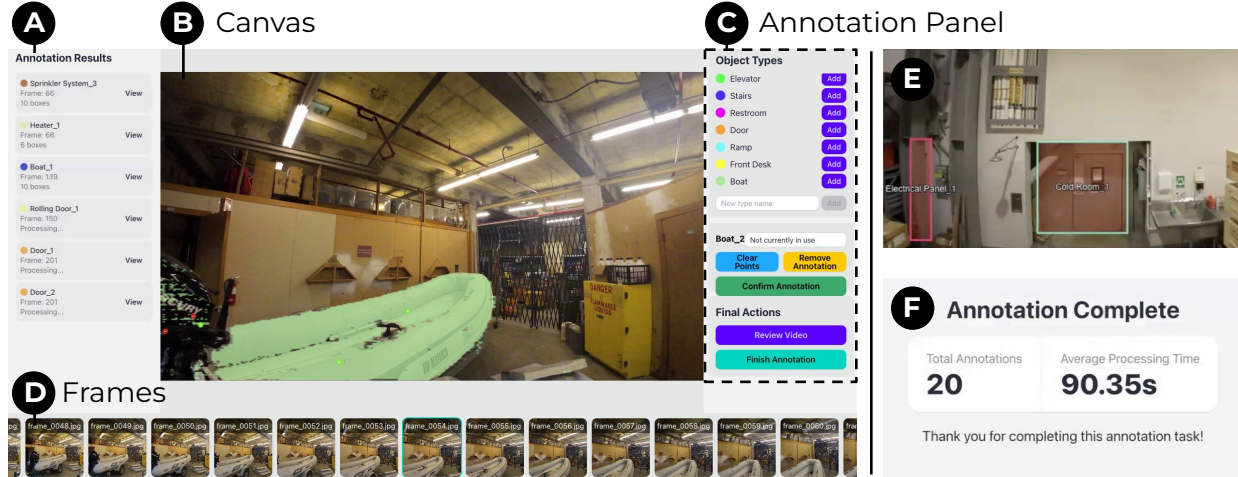


Figure 4.8: The annotation interface. (A) The results panel lists confirmed annotations. (B) The canvas displays selected frames and supports click-based segmentation. (C) The annotation panel provides object types and commands such as “Clear Points,” “Confirm Annotation,” and “Finish Annotation.” (D) The frames panel lists available video frames. (E) After the user confirms an annotation, the server propagates its segmentation mask to subsequent frames, where results appear as bounding boxes. (F) The interface summarizes the completed annotations.

reliably. Second, the function and importance of a facility depend on site-specific practices and stakeholder needs. We therefore designed an annotation interface for collaboratively creating POIs in reconstructed 3D maps and a review interface for inspecting the resulting interactive models.

4.4.3.1 Annotation Interface

We created a web interface for efficient indoor POI annotation. Unlike traditional indoor-mapping workflows that rely on CAD files and specialized design skills, the interface presents frames from drone footage and lets users mark target objects, such as entrances, doors, and stairs, with clicks. The interface has four main components (Figure 4.8): an annotation canvas that displays frames, points, masks, and processed bounding boxes (B); a frames panel that lists the video frames chronologically (D); an annotation-tools panel that provides object types and task commands (C); and a results panel that lists confirmed annotations, reports their processing status, and supports review (A).

Users browse the frames panel and select a frame to annotate. They can create an object instance by choosing a predefined type or defining a new one, optionally adding a detailed description. After the user clicks the target object, an embedded SAM2 model [Ravi et al., 2024a] produces an interactive segmentation

mask. Positive clicks add unmasked regions, whereas negative clicks remove masked regions. Users may clear the points, delete the instance, or confirm the annotation. Confirmation sends the mask to the server, where SAM2 propagates it across subsequent video frames. The interface displays the propagated results as bounding boxes (Figure 4.8E), and the ray-casting pipeline converts them into 3D bounding boxes in the reconstructed mesh (subsubsection 4.4.2.4).

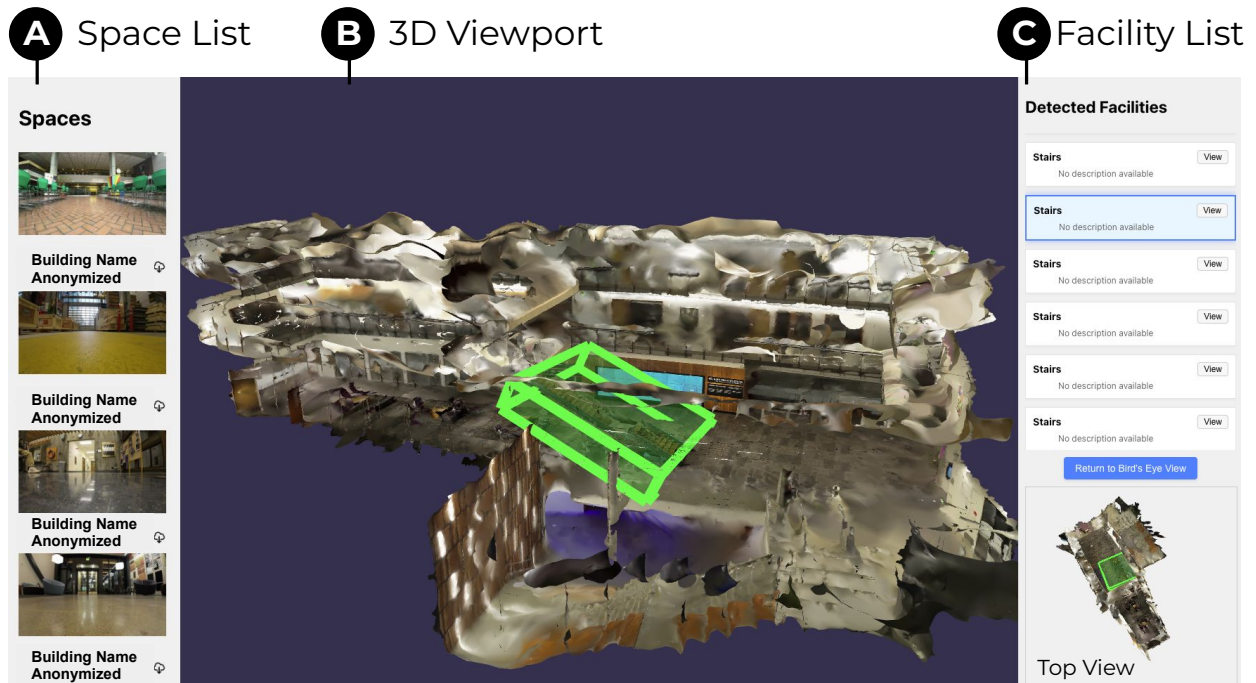


Figure 4.9: The review interface. (A) A list of available spaces. (B) A 3D viewport displaying the map and annotated objects. (C) A list of annotated facilities that users can highlight or select with the “View” button. Additional controls return the camera to a bird’s-eye view or display an orthographic floor-plan view.

4.4.3.2 Visualization Interface

We also implemented a web interface for reviewing the reconstructed mesh and the 3D bounding boxes that mark annotated objects (Figure 4.9). The interface contains a list of available maps (A), an interactive 3D canvas (B), a facility list (C), and a top-down orthographic view that presents the model as a floor plan. Users can drag to rotate the camera, browse and highlight objects, and select an object to focus the view on its bounding box.

4.4.4 Evaluation

We evaluated FlyMeThrough’s technical performance and overall usability. We collected drone footage from 12 spaces that varied in type, size, ceiling height, and function, then processed the footage into 3D indoor maps to assess the pipeline’s technical capabilities. We evaluated the interface with 10 participants, including five building managers and five building occupants, to examine usability, practical implications, and potential barriers to deployment.

The FlyMeThrough human-subjects study was reviewed and approved by the University of Washington Institutional Review Board under the project’s prior name, DIAM (UW IRB; STUDY00022431).

4.4.4.1 Procedure

Before conducting indoor drone flights, we asked building managers to authorize and supervise flights in their buildings. With their approval, the lead author manually operated the drone during off-hours to minimize risk and disruption. We collected footage from 12 indoor spaces of varying sizes, types, and functions, then reconstructed the spaces to test the pipeline’s robustness (Table 4.2).

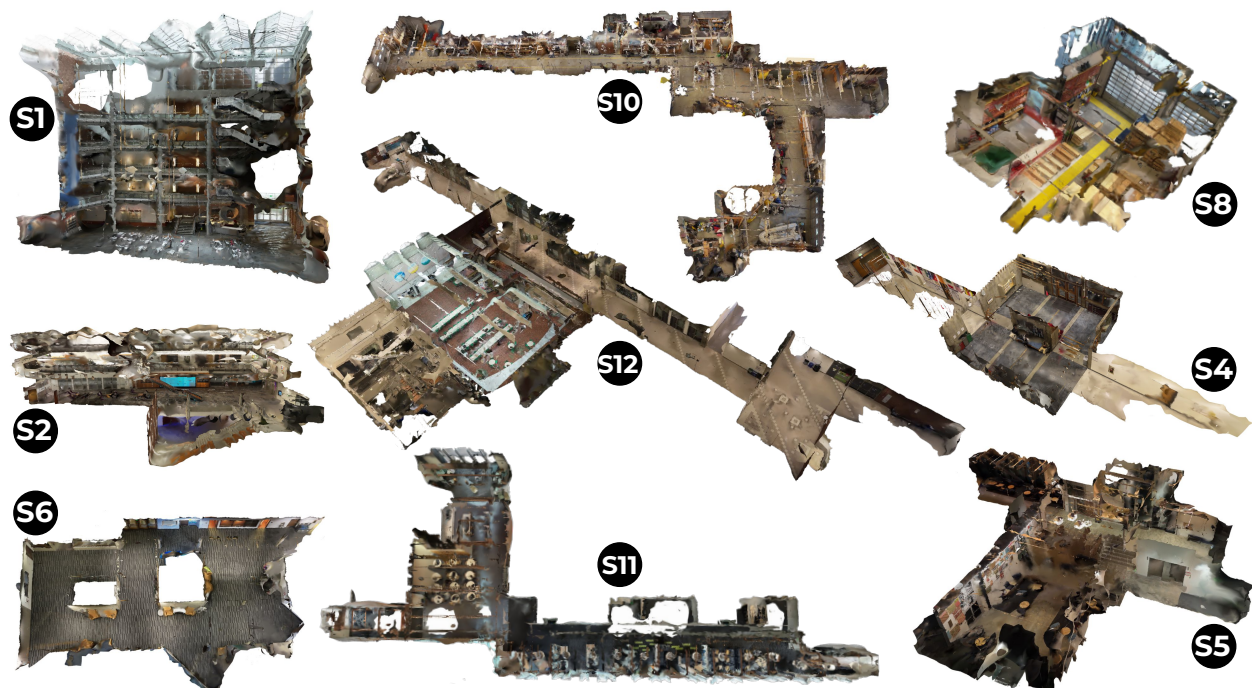


Figure 4.10: Nine of the scanned spaces.

Table 4.2: Overview of spaces, video data, and reconstruction status.

Space ID	Function	Area (m ²)	Height (floors)	Video duration (min:s)	Reconstruction status
S1	Education	1600	6	3:45	Good
S2	Education	1400	4	2:33	Good
S3	Education	100	1	2:08	Good
S4	Office	120	1	3:22	Good
S5	Education	180	1	2:10	Good
S6	Education	600	1	3:06	Good
S7	Exhibition	3000	2	6:07	Good
S8	Engineering	260	3	5:03	Good
S9	Office	1100	4	5:33	Failed
S10	Engineering	2000	1	5:53	Good
S11	Education	800	1	4:02	Good
S12	Education	2400	3	9:13	Good

To evaluate the interface, we conducted remote and in-person study sessions in which participants annotated POIs and reviewed 3D maps. Each session had five sequential parts: (1) initial setup, (2) a drone-mapping demonstration, (3) use of the annotation interface, (4) an interview and discussion, and (5) review of the final model and concluding feedback. Each session lasted around 60 minutes.

The session began with informed consent and a brief background survey. Building managers received additional questions about their current building-evaluation practices. Participants who had not witnessed a drone flight watched recordings from both the drone’s and a bystander’s perspectives. This dual-view presentation helped them understand the drone’s operational capabilities and physical presence in the space. Participants then used the annotation interface. We asked open-ended questions about their reactions to the flight process, the interface’s usability, and possible integration into existing building-evaluation practices. While participants discussed these questions, the system processed their annotations into an instance-enriched 3D model. Participants reviewed the completed model through the interactive interface and answered concluding questions about the system and its relevance to their needs.

4.4.4.2 Participants

We recruited five building managers who supported the drone flights and participated in the user study, along with five building occupants who regularly used the scanned buildings. Four participants witnessed drone operations in person; the remaining participants watched recordings. Participants received \$25 per hour

for guiding or observing flights and participating in the study. Table 4.3 provides additional demographic information.

Table 4.3: Participant demographics and annotation performance.

Participant	Age	Gender	Role	Space tested	Experience (years)	Footage/witness	Annotations	Success (%)
M1	35-44	M	Building Manager	S1	13	Footage	12	100.00
M2	45-54	M	Building Manager	S8, S10	7	Witness	33	75.76
M3	25-34	M	Building Manager	S2	9	Footage	19	100.00
M4	18-24	F	Building Manager	S11	1.5	Witness	16	87.50
M5	55+	M	Building Manager	S12	7	Footage	9	100.00
O1	25-34	M	Occupant	S6	5	Witness	29	93.10
O2	25-34	M	Occupant	S2	5	Footage	8	87.50
O3	25-34	M	Occupant	S4	3.5	Witness	8	100.00
O4	25-34	M	Occupant	S5	0.5	Footage	14	100.00
O5	25-34	F	Occupant	S1	6	Footage	10	100.00

4.4.4.3 Analysis Approach

Our analysis of the semi-structured interviews focused on summarizing high-level themes. One researcher developed a set of themes through qualitative open coding [Charmaz, 2006] based on the video transcript, then coded the responses according to the themes. Participant quotes have been slightly modified for concision, grammar, and anonymity.

The system logged all annotations, segmentations, and subsequent ray-casting results. The research team analyzed these data for computational efficiency and robustness.

4.4.4.4 Technical Results

Reconstruction succeeded in 11 of the 12 target spaces. The only failure occurred in S9 (Table 4.2; Figure 4.11A). Figure 4.10 shows nine successful reconstructions. During the study sessions, participants created an average of 15.8 annotations per space (Table 4.3). SAM2 processed every annotation, taking an average of 41.10 seconds per annotation ($SD = 11.21$). Each ray-casting operation took an average of 63.6 seconds ($SD = 48.65$), and 91.77% of annotations were successfully converted into bounding boxes for review. Table 4.3 summarizes annotation counts and casting success rates by participant.

We also analyzed the failed casting cases. A total of 13 failures were recorded, of which 8 (61.5%) were due to missing camera parameters in the corresponding frames at the beginning of the SfM-based indoor

reconstruction process. The remaining five failures (38.5%) resulted from incomplete SfM reconstructions and missing geometry (Figure 4.11B and C).

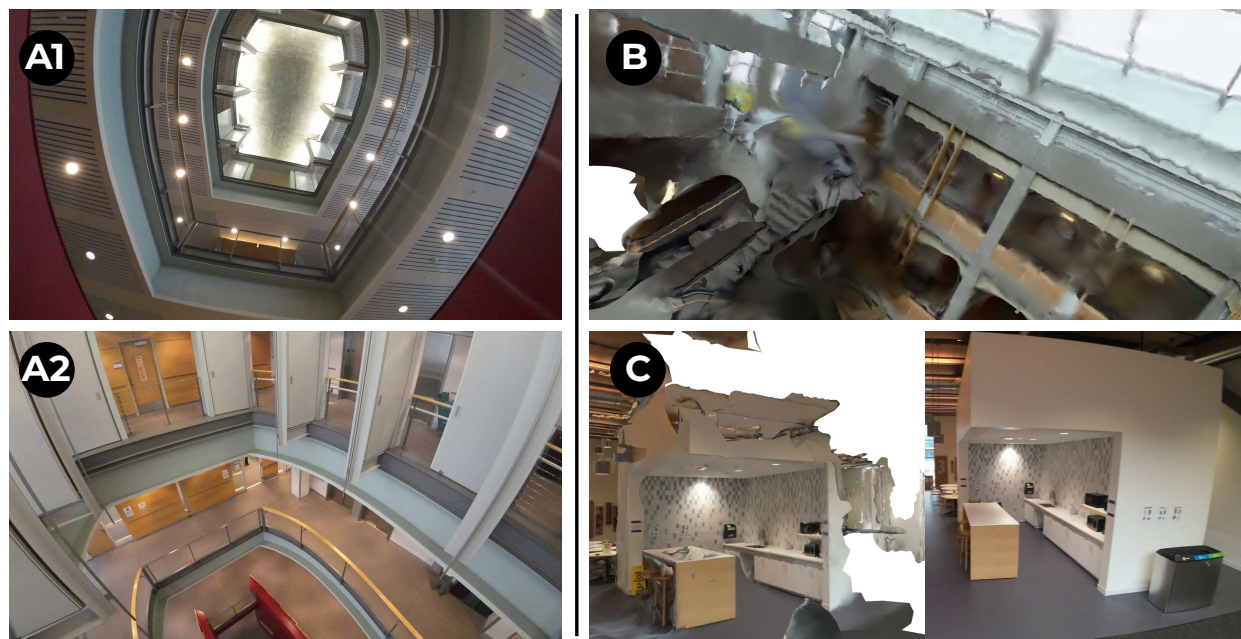


Figure 4.11: Three reconstruction failure modes. (A) Two images from S9, a multilevel office atrium whose reconstruction failed. (B) Blurring in a sparsely captured region of a reconstructed map. (C) Missing reconstructed geometry for a waste container visible in the source footage.

4.4.4.5 Findings

A total of 158 objects were annotated by the participants. Among them, doors (including regular and rolling doors) were the most frequently labeled (31), followed by common building infrastructure such as stairs (21), entrances (10), elevators (9), restrooms (5), and ramps (3). In addition to these core architectural features, participants also annotated a wide range of smaller objects (e.g., signage, furniture, equipment). We also observed a distinction between building managers and occupants: managers tended to label technical and safety-related features, such as fire-control systems and equipment panels, whereas occupants more frequently annotated everyday elements such as sofas, whiteboards, and waste bins.

Below we summarize the major themes that emerged from our user interviews.

Annotation-System Usability and Areas for Improvement. Our evaluation of FlyMeThrough’s annotation interface revealed generally positive feedback regarding its usability. When asked to rate ease of use on a 7-point Likert scale (1=very difficult, 7=very easy), participants reported a median rating of 6 ($IQR=2.00$), indicating an overall favorable assessment of the system’s usability. Several participants explicitly characterized the interface as intuitive, with building managers and occupants noting it was “*pretty easy to use*” (M2, O5) and “*not complicated at all*” (M5). This suggests that FlyMeThrough’s annotation workflow can accommodate users with varying levels of technical expertise.

Despite the generally positive reception, we also identified areas for improvement. A common challenge involved the object-selection workflow: “*The confusing thing is you have to click this first and then identify the thing again... the order is maybe a little bit confusing*” (M1). Similarly, M4 noted: “*Sometimes I would forget to [select the object type first]. I would click on an object and then realize I didn’t select what type of object it was.*” Participants also expressed interest in additional functionality that could enhance their experience. For example, O2 suggested implementing an example-based approach: “*I think for the annotation part, it might be helpful if I can be provided with examples so I don’t have to start on my own. I can just say yes or no.*” This comment points to the potential benefit of combining automated annotation with human verification.

Perceived Model Quality. When asked to rate the quality of the model on a scale of 1-7 (1=poor, 7=excellent), participants had a median rating of 5 ($IQR=0.75$), indicating generally positive perceptions of the system’s output quality. We also asked participants to give a binary rating (accurate/inaccurate) to the location and size of the bounding box for each of the annotated objects. The average percentage of objects rated as accurate was 71.54% ($SD=29.73\%$). This wide standard deviation reflects considerable variation in how participants perceived the annotation results.

Some participants found the model’s output to be satisfactory and accurate for their needs. However, others expected more precise geometric alignment, particularly with architectural elements. As one building manager explained: “*so you end up with a larger bounding box and it’s at an angle relative to where you’re viewing the individual item, as opposed to the item itself being a rectangular prism up in the ceiling*” (M2). Similarly, M5 expressed that he was hoping the bounding box would exactly fit the volume of the room: “*the box doesn’t line up with this atrium exactly*”. This contrast also reflected the varied scale of labeled entities,

which ranged from entire atria to small electrical panels. These findings suggest that while the current model quality is sufficient for many use cases, expectations for geometric precision vary considerably across users and object types.

Concerns About Indoor Drone Flights. When asked about indoor drone flights, participants reported few safety concerns. They focused primarily on appropriate scanning locations and privacy. Participants suggested conducting scans during off-hours and limiting them to public areas: *“common spaces would be [appropriate to scan]. Like we could do the hallways, we could do the large atria”* (M1). Multiple building managers (M1, M2, M3) emphasized avoiding sensitive spaces such as deans’ offices, research labs, and dormitories. Privacy emerged as a key consideration. M4 recommended providing advance notice to building tenants before scanning so individuals could avoid the area if they did not wish to be included. Building occupants raised concerns about facial privacy, with one suggesting that *“maybe it’s best to blur people’s faces before annotation”* (O5).

Diverse Use Cases and Stakeholder Benefits. Participants demonstrated diverse needs that could be fulfilled by FlyMeThrough. Building managers and occupants identified distinct use cases, with several participants recognizing potential benefits beyond their own user group.

For Building Managers. Building managers identified numerous practical applications for FlyMeThrough, ranging from daily operational tasks to strategic space planning and resource management. Our interviews revealed that FlyMeThrough has strong potential to enhance current mapping practices. One manager highlighted how the system could eliminate redundant site visits: *“Sometimes I’ll go look at a room or space and then if I don’t take a picture I go back to my office. If I forget then I just have to go back. But it’s nice to have all the data right here”* (M1). This suggests that having comprehensive spatial data readily available could improve workflow efficiency. The system’s potential for remote inspection was particularly valued for maintenance operations. M1 explained: *“It just saves us from having to go check on things, for example, there’s a leak from the roof”*. Another specifically highlighted specialized applications: *“This would come in most handy from a facilities perspective if we did the 3D mapping in a mechanical room, where all the pumps and valves, piping and stuff like that”* (M5). These comments suggest that the system could reduce the need for in-person inspections of routine issues and provide valuable documentation of complex

mechanical spaces.

Beyond maintenance, managers saw potential for enhancing event planning and coordination. One manager shared: *“All the images I have of the atrium are just me taking pictures with my phone from different angles. But it could be really helpful to just have this. Then for example, if I have an event coming in, I could say to them, here’s the space. We could mount lights here on this column and this column”* (M1). Training applications were also identified as valuable use cases. As M3 noted: *“If for whatever reason I cannot meet with somebody for a health and safety training, I could do it virtually with them or for them to be able to review that space after the safety training as a records”* (M3). This points to potential educational applications beyond basic navigation and space management.

At a more strategic level, managers recognized the system’s potential for space utilization analysis. As M2 noted: *“What’s the square footage that we’re using for storage versus classrooms versus offices versus labs? Those are things that are actually important to the university from a sense of when grants go in and charging rates to grants for infrastructure”* (M2). This highlights FlyMeThrough’s potential to support not only operational decision-making but also financial planning and resource allocation.

For Building Occupants. Building occupants focused primarily on navigation applications. One noted potential use in large, complex spaces: *“For conference, if you go to convention center, it’s super huge. Also for direction”* (O2). The same participant highlighted accessibility benefits: *“Maybe if I went to a new country that I don’t speak their language, but I want to go to a museum. I want to see a specific artifacts, I want to go to a specific section. I can just use this. Not reading anything”* (O2). This suggests that visual-based navigation could overcome language barriers in unfamiliar environments.

Essential amenities and safety information were prioritized by occupants: *“When I go to a new building, I always want to know the emergency exit and also bathrooms and water. Those are the most important thing. And whether there are elevators or stairs to go to the other place”* (O5). This emphasizes the importance of including basic amenities and safety features in the annotation system.

Cross-User Benefits Notably, many participants recognized potential benefits beyond their immediate needs. One manager articulated how the system could serve multiple stakeholders: *“It would be the kind of space that would actually work really well for people at all level of the department. Grad students could see their office before they move in. Facilities could get a chance to look at a space without having to spend*

30 minutes getting a car and driving across campus. It could work in group meetings talking about space allocation, you could pull this up and have a conversation in a meeting space as opposed to having to walk through it. So everybody that uses the space could find a way to benefit from it” (M2). This perspective suggests that FlyMeThrough has the potential to serve as a unified spatial information platform that bridges the needs of different stakeholder groups, from administrators and facilities staff to everyday users and visitors.

4.4.5 Discussion and Future Work

In this project, we introduce FlyMeThrough, a drone-based indoor-mapping system that uses commodity drones to create interactive, POI-enriched 3D maps. A technical evaluation across 12 indoor spaces and a study with 10 participants demonstrated the system’s feasibility and usability while revealing potential applications and opportunities for improvement.

4.4.5.1 Application Scenarios

As discussed in the evaluation, FlyMeThrough supports application scenarios in which building managers, occupants, and first-time visitors inspect indoor spaces. The tool can support navigation, room location, and identification of features ranging from doors and stairs to electrical panels and automated external defibrillators. Because its annotations are customizable, users can create different maps for different needs. The system also represents small, movable, or temporary structures, such as furniture, cargo boxes, and exhibition installations, that traditional architectural maps often omit. This capability could support event planning, exhibitions, and furniture logistics. The high-resolution 3D maps may also provide spatial cues that help people assess accessibility before visiting, although this potential requires evaluation with the relevant disability communities.

4.4.5.2 End-User Interface

At this stage, we have implemented a web interface (Figure 4.9) to showcase the final 3D map. In future work, we plan to develop additional interface versions tailored to different user needs. For example, a navigation interface could better support spatial movement and route planning to specific indoor facilities;

a facility management interface could focus on accessing and managing building safety information and equipment; and a mobile interface could support real-time user localization by matching camera feeds with the 3D indoor map to enable live navigation assistance. We also envision accessibility-focused interfaces that evaluate and visualize spatial accessibility based on individual user requirements. Informed by our user study findings and the rich capabilities of our 3D mapping system, there is significant potential to explore a wide design space for future 3D mapping interfaces.

4.4.5.3 Improving Model Quality and Reconstruction Efficiency

As discussed in the user study and limitations, the current 3D models generated using the commercial photogrammetry tool MetaShape [Over et al., 2021] still leave room for improvement in terms of quality. In future work, we aim to explore alternative 3D reconstruction methods that are fully open-source, better automated, and capable of producing higher-quality 3D maps.

Among the methods evaluated, MAST3R-SLAM [Murai et al., 2024], a SLAM-based approach built on recent 3D reconstruction foundation models [Wang et al., 2024b; Leroy et al., 2024], has shown promising results with relatively modest computational requirements. Our current SfM pipeline in MetaShape requires several hours of computation per scene, whereas MAST3R-SLAM combines feed-forward foundation models with efficient SLAM processing to complete reconstructions in minutes. Although its reconstruction quality does not yet match MetaShape, rapid progress in 3D foundation models makes this a promising direction.

Another promising direction is generative view synthesis [Weber et al., 2025], which enables the synthesis of novel views from captured images. This capability can help enhance reconstruction quality in areas that are occluded or otherwise inaccessible to the drone, and potentially reduce the extent of capture required.

We plan to continue exploring these open-source alternatives and will consider transitioning to them as their performance becomes suitable for our use cases.

4.4.5.4 Automating Drone Flights

Future work should investigate regular indoor data collection through automated drone flights. Prior work has explored autonomous path planning for drones in indoor environments [Gao et al., 2023; Li et al., 2018; Zhang et al., 2024; Maboudi et al., 2023; Gao et al., 2024], providing a foundation for software-level automation on commodity hardware. Participants reported few concerns about the supervised flights used in this study, but autonomous collection introduces additional questions about bystander consent, capture timing, coverage, and obstacle avoidance. No-fly zones and congestion detection could reduce risk, but they require evaluation before deployment.

4.4.6 Conclusion

In this project, we present FlyMeThrough, a drone-based indoor-mapping system that uses RGB-only footage and human-AI collaborative annotation to generate POI-enriched 3D maps. We evaluated the system across 12 indoor spaces of varying sizes, functions, and layouts. A study with building managers and occupants found generally positive perceptions of the interface’s usability, while the technical evaluation established the pipeline’s feasibility and exposed reconstruction and localization failures. Participants also proposed applications spanning building management, remote inspection, and navigation.

4.5 Frontier Synthesis

This frontier extends spatial perception across places and times that a person cannot inspect directly. RAIS and FlyMeThrough show that doing so requires more than replacing a person with a mobile sensor. A useful capture system needs a task model, a coverage strategy, and a representation that preserves functional meaning. RAIS begins with a specified accessibility rubric and automates the scan needed to evaluate it. FlyMeThrough begins with a more varied building-management setting and uses human annotation to determine which facilities should become persistent spatial entities.

Together, the projects reveal a continuum of automation and uncertainty. At one end, the task and required evidence are sufficiently structured that a robot can execute a largely predefined scan. At the other, the environment and stakeholder needs are too varied for complete automation, but AI can greatly amplify

sparse human input. In both cases, the final model is a shared spatial representation rather than a static copy of the world. It records what the sensing platform could capture, what the task definition or annotator selected, and what the reconstruction pipeline could reliably localize. Remote users need access to these provenance and uncertainty boundaries if they are to make responsible decisions from the model.

Chapter 5

Frontier III: Aligning AI Spatial Reasoning with Human Needs

5.1 Preamble

The third frontier concerns alignment between an AI system’s spatial grounding and predictions and the conditions under which a particular person or embodied agent can act. Such alignment requires more than semantic and topological understanding of the environment. It requires translating physical evidence into judgments that respect an agent’s capabilities, goals, preferences, and the consequences of being wrong.

This issue matters whenever spatial AI informs action. A description can be factually plausible yet practically unusable if it assumes the wrong subject condition. In accessibility contexts, treating a generic standard as a complete account of a person’s needs can erase consequential variation; personalized assessment and shared-control research instead foreground the need to negotiate individual criteria and preserve the user’s ability to monitor and intervene [Huang et al., 2025; Kamikubo et al., 2025].

Capability-conditioned reasoning is difficult because the relevant evidence is distributed across semantics, scale, topology, and function. A system may need to combine observations from several viewpoints, estimate a metric relation rather than recognize an object category, and reason about alternatives under incomplete information. Recent spatial-reasoning benchmarks show that multimodal models remain unreliable on multi-view and metric judgments and generalize unevenly from constrained indoor settings to

dynamic, open-world environments [Yang et al., 2025b; Wu et al., 2026]. Even a correct physical measurement does not determine feasibility without an explicit account of the agent, task, and uncertainty in the evidence.

In this chapter, we take indoor navigation as a focused test case for this general human-AI spatial alignment problem. Navigation requires a system to reason over routes rather than isolated objects, while feasibility can change sharply when a traveler or robot has different constraints. Existing benchmarks commonly assume a standard agent or assess arrival under fixed simulator conditions, allowing strong average performance to conceal failures that emerge only when the same environment is interpreted for different embodiments. CapNav develops this test case as a capability-conditioned indoor-navigation benchmark. By holding the environment and task constant while varying the agent profile, it examines whether models change their spatial judgments when the conditions for action change.

5.2 CapNav: Capability-Conditioned Indoor Navigation

CapNav is a benchmark for evaluating vision-language models on indoor navigation under human and robotic capability constraints. Each query combines a tour video, a semantic navigation graph, a start-to-goal task, and one of five capability profiles. Evaluation decomposes performance into feasibility classification, path validity, route traversability, and reasoning validity. This decomposition makes it possible to diagnose whether a model recognized that a task was possible, actually produced a legal route, respected every edge-level constraint, and supplied a valid reason for infeasibility.

5.2.1 Introduction

As vision-language models (VLMs) advance in visual grounding [Shi et al., 2024] and spatial reasoning [Cheng et al., 2024], they are increasingly used as navigation assistants or planning modules for robots and assistive systems [Shah and et al., 2023; Zhou et al., 2023; Goetting et al., 2024]. They guide people to destinations [Li et al., 2025; Goetting et al., 2024] and support robot movement decisions [Huang et al., 2022] in open-ended scenarios where multiple routes may be valid and feasibility depends on movement capabilities.

Consider the multistory building shown in Figure 5.2. To travel from the basement foyer to the top floor,

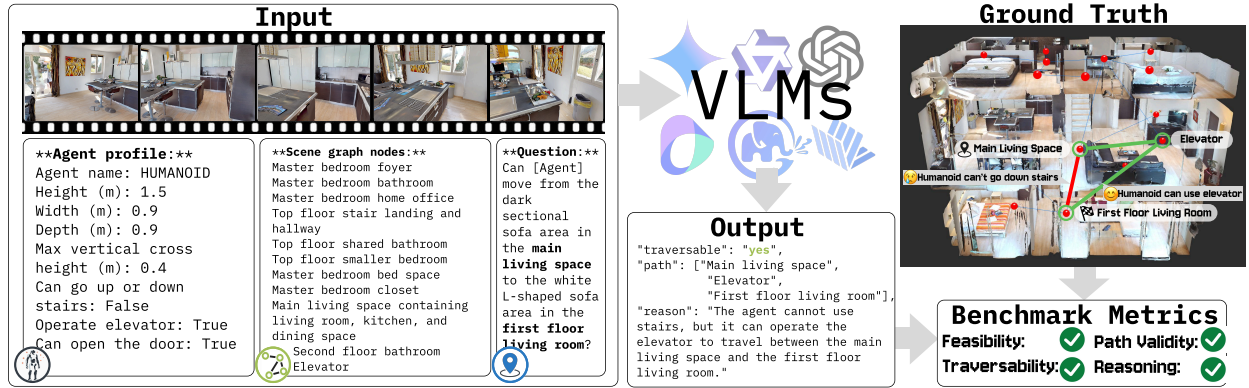


Figure 5.1: CapNav evaluates capability-conditioned navigation from a tour video, navigation-graph nodes, an agent profile, and a navigation task. Outputs are assessed for feasibility, path validity, route traversability, and reasoning validity.

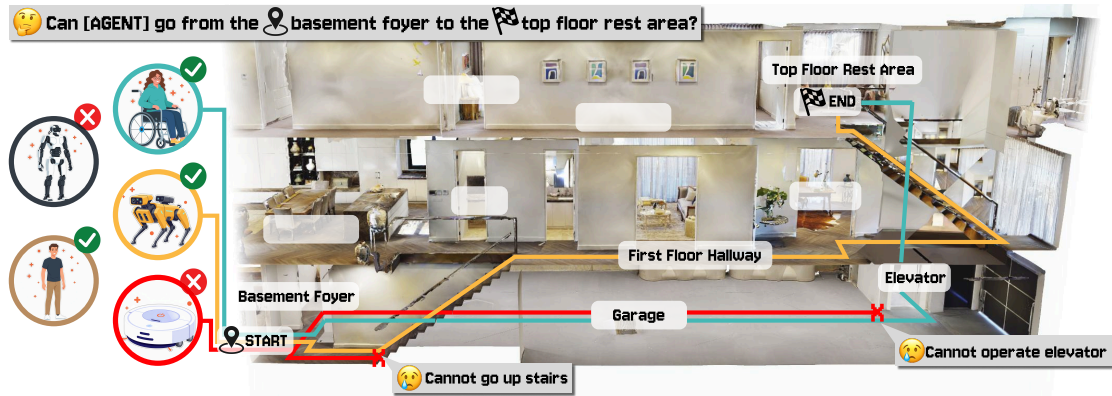


Figure 5.2: The CapNav benchmark evaluates whether VLMs account for differences in agent mobility when generating navigation plans. In this example, route feasibility and the appropriate path differ across agents.

a wheelchair user must use the elevator, whereas a quadrupedal robot must use the stairs because it cannot operate the elevator. Similarly, an adult without motor disabilities may pass through a cluttered corridor that is too narrow for a humanoid robot. Without capability information, a VLM may recommend an infeasible or unsafe route [Li et al., 2023; Guan et al., 2024].

Such mobility constraints and the plurality of valid navigation solutions remain underexplored in prior vision-language navigation research. Existing frameworks and benchmarks evaluate embodiment-agnostic goal achievement in simulated or graph-based environments [Anderson and et al., 2018b; Krantz and et al., 2020; Wang et al., 2025b; Song et al., 2025] or reduce navigation to simplified VQA-style spatial-reasoning tasks [Feng et al., 2025; Wu et al., 2025; Yang et al., 2025b; Xu et al., 2025]. Moreover, evaluation protocols typically measure trajectory fidelity against a single annotated ground-truth path [Jain et al., 2019; Ilharco and et al., 2019; Anderson and et al., 2018a]. Consequently, these paradigms overlook capability-conditioned validity and the multiple feasible solutions that characterize real-world navigation. This limitation becomes increasingly consequential as VLMs are deployed in embodied-control and assistive scenarios.

We introduce CapNav, a benchmark for VLM navigation validity under varying mobility constraints. We consider five representative embodiments: adults without motor disabilities, wheelchair users, sweeping robots, humanoid robots, and quadrupedal robots. We use real-world 3D scans [Chang et al., 2017; Ramakrishnan and et al., 2021] and annotated navigation graphs [Anderson and et al., 2018b; Savva et al., 2019; Krantz and et al., 2020] to support multiple valid solutions and per-edge traversability analysis. CapNav takes an agent description, realistic indoor video, key spatial reference nodes, and a navigation task from A to B. It then evaluates VLM predictions of feasibility, navigation path, and reasoning validity. In total, CapNav contains 45 scenes, 2,365 navigation tasks, and 5,075 ground-truth traversability annotations.

Using the CapNav benchmark, we evaluated 13 open-source and proprietary VLMs under varying inference settings. Quantitative and qualitative analyses revealed three major patterns: (1) **mobility degradation**: navigation performance decreases when mobility constraints are introduced; (2) **visual bottleneck**: increasing the visual-input budget does not consistently improve CapNav performance; and (3) **dimension neglect**: all tested models struggle to detect nontraversability caused by insufficient spatial clearance, such as narrow passages or limited turning space.

These results show that substantial progress is still needed before VLM-based navigation can reliably

support embodiment-specific deployment, particularly in safety-critical scenarios. Current VLM architectures and training paradigms remain limited in their ability to integrate spatial information across multiple visual frames and perform precise metric reasoning in complex environments. Advancing embodied navigation therefore requires geometry-grounded reasoning and training objectives that explicitly encode mobility constraints and metric dimensions.

Our contributions are threefold:

1. **The CapNav benchmark.** We introduce CAPNAV, a capability-conditioned VLN benchmark that evaluates VLM navigation in complex indoor environments under mobility constraints across five representative human and robot embodiments.
2. **Comprehensive evaluation of VLMs.** We conduct a comparative assessment of 13 state-of-the-art VLMs on capability-aware feasibility prediction, path validity, route traversability, and reasoning quality.
3. **Guidelines and resources.** We analyze the effects of visual-frame budgets, model settings, obstacle types, and failure modes. We also release the videos, tasks, agent profiles, 5,075 traversability annotations, and an interactive annotation interface for extending CapNav.

5.2.2 The CapNav Benchmark

We introduce capability-conditioned navigation (CapNav), a benchmark for assessing VLMs’ ability to navigate built environments under specific physical capabilities.

5.2.2.1 Problem Statement

In CapNav, each query instance is defined by a *space-task-capability* triple $\langle \mathcal{S}, \tau, \mathbf{a} \rangle$ where the space $\mathcal{S}$ is represented by a touring video and a set of key spatial nodes; τ is a natural-language navigation goal specifying the source and target locations; and $\mathbf{a}$ is an agent profile encoding physical dimensions and operational abilities. Given $\langle \mathcal{S}, \tau, \mathbf{a} \rangle$, a vision-language model produces

$$(\hat{y}, \hat{P}, \hat{\rho}) = f_{\theta}(\mathcal{S}, \tau, \mathbf{a}),$$

where $\hat{y} \in \{0, 1\}$ denotes task feasibility (CAN/CANNOT complete), $\hat{P} = [v_0, \dots, v_m]$ is a node sequence representing the proposed navigation path, and $\hat{\rho}$ is a concise rationale explaining infeasibility when $\hat{y} = 0$.

This formulation enables CapNav to evaluate capability-aware navigation from four complementary aspects: (i) navigation feasibility, (ii) path validity, (iii) traversability accuracy, and (iv) reasoning quality for infeasible cases.

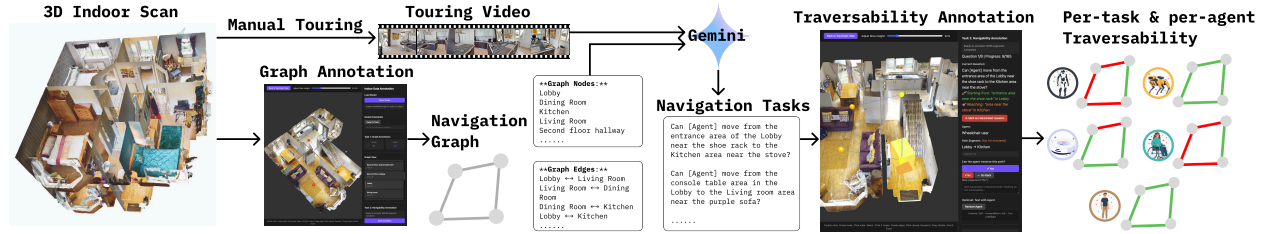


Figure 5.3: Overview of CapNav data construction. Starting from a 3D indoor scan, we manually record a touring video and construct a navigation graph. We then use Gemini to generate natural-language navigation tasks. Finally, we annotate traversability for each task and agent by manually controlling virtual agents in the annotation interface.

Space. Each space $\mathcal{S}$ is a scanned 3D indoor environment comprising multiple interconnected rooms. We represent its spatial information through a rendered tour video. To ground navigation in language, we manually create a connectivity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose nodes represent spatial regions (e.g., *kitchen*, *entry foyer*, and *hallway*) and whose edges represent direct, walkable connections.

Task. A navigation task τ is a natural-language instruction specifying a movement goal from a source node to a target node, for example, “*From the wooden cabinet in the entrance foyer, go to the desk area in the master bedroom.*” Formally, each goal τ can be expressed as a directed pair:

$$\tau : (v_{\text{src}}, d_{\text{src}}) \rightarrow (v_{\text{tgt}}, d_{\text{tgt}}), \quad v_{\text{src}}, v_{\text{tgt}} \in \mathcal{V},$$

where v_{src} and v_{tgt} denote the source and target nodes in $\mathcal{V}$, and d_{src} and d_{tgt} are their corresponding natural-language descriptions specifying precise starting and ending positions within those nodes, for example, “*at the wooden cabinet in the entrance foyer*” and “*beside the desk area in the master bedroom.*” This representation captures both the high-level spatial nodes and the finer-grained textual localization needed for realistic indoor navigation reasoning.

Agent Profiles. We specify mobility capabilities $\mathbf{a}$ with five representative agent profiles covering human mobility and common robot platforms: adults without motor disabilities, for whom all indoor routes are assumed achievable; wheelchair users, who cannot use stairs and require clearance for passage and turning; humanoid robots, which cannot climb stairs and require passage clearance; sweeping robots, which require a flat floor surface; and quadrupedal robots, which can traverse most indoor spaces but cannot operate doors or elevators.

Each profile is described by a capability JSON object, $\mathbf{a} = (\phi, \kappa, \mu)$, where ϕ captures the physical footprint; κ captures vertical-traversal limits, including the maximum height of a single traversable obstacle and whether the agent can climb a flight of stairs; and μ captures operational and manipulation abilities. These attributes affect both edge-level traversability in $\mathcal{E}$ and overall task feasibility for τ . Figure 5.4 shows examples.

5.2.2.2 Traversability Ground Truth

We manually annotate traversability for each task and agent type. Binary edge-level labels allow multiple possible routes and provide precise evidence when a route is not traversable. For each navigation task τ and embodiment $\mathbf{a}$, annotators review every simple path that connects v_{src} and v_{tgt} in $\mathcal{G}$ and assign $g_e^{(\mathbf{a})} \in \{0, 1\}$ to each edge e on these paths, indicating whether the edge is passable for $\mathbf{a}$ given the local geometry and mobility capabilities. Annotators provide a textual reason for each nontraversable edge, such as “cannot go up or down stairs.” The annotation interface visualizes 3D colliders that match ϕ and supports manually controlled movement and turning to verify clearance and maneuvering space (Figure 5.3).

Given these edge-level labels, we define the feasibility ground truth for a task $(v_{\text{src}}, v_{\text{tgt}})$ as the existence of at least one simple path (an acyclic sequence of distinct nodes) $P(v_{\text{src}}, v_{\text{tgt}})$ in $\mathcal{G}$ whose edges are all traversable:

$$y^* = \mathbb{I}[\exists P(v_{\text{src}}, v_{\text{tgt}}) : \forall (u, v) \in P, g_{(u,v)}^{(\mathbf{a})} = 1].$$

5.2.2.3 Dataset

We collected 3D indoor scenes from HM3D [Ramakrishnan and et al., 2021] and Matterport3D [Chang et al., 2017] and recorded tour videos by manually navigating each space in the Habitat simulator [Savva

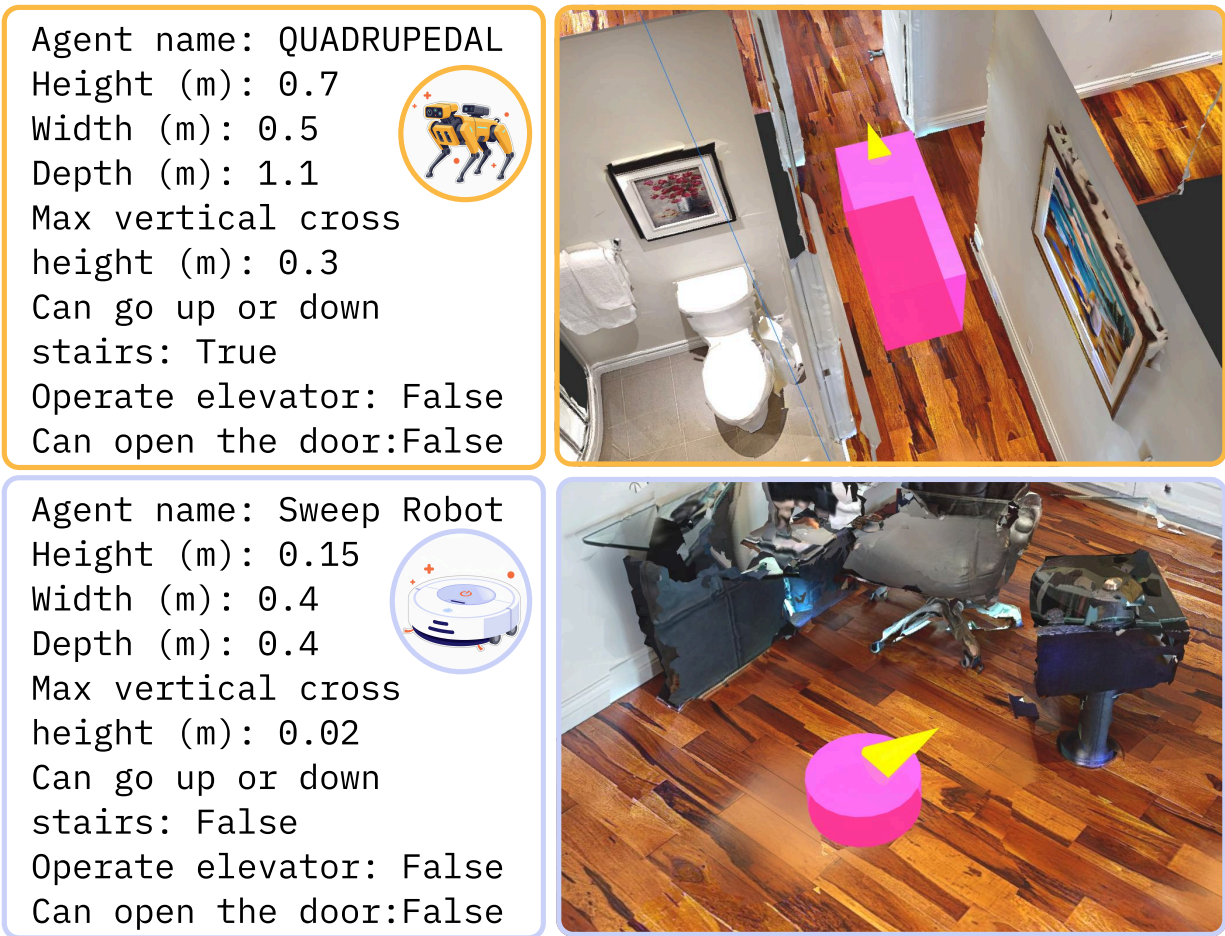


Figure 5.4: Examples of CapNav agent profiles. Left: each profile specifies physical dimensions and functional capabilities. Right: virtual agents with the specified dimensions are rendered and maneuvered in the 3D scenes to help verify traversability.

et al., 2019]. We rendered the videos at 2 FPS from a height of 1.5 m with a 75° field of view to approximate a casual handheld walkthrough. For each scene, we constructed a navigation graph with a custom annotation interface. Nodes $v \in \mathcal{V}$ have semantic labels $c(v)$ and approximate 3D positions, and edges $e = (u, v) \in \mathcal{E}$ denote bidirectional, directly walkable connections. We provided each scene’s video and node list to Gemini 2.5 Pro to generate navigation tasks, manually verified each task, and annotated edge-level traversability for the five embodiments, including reasons for nontraversable cases. After excluding incomplete scenes, disconnected subspaces, repetitive semantics, and trivial layouts, we retained 45 indoor scenes. Videos had a mean duration of 160.38 seconds, and scenes had a mean of 13.8 nodes and 14.5 edges. The resulting dataset contains 2,365 navigation tasks and 5,075 traversability labels, of which 3,945 are positive and 1,130 are negative. Task feasibility and edge traversability vary across agent types (Table 5.1).

Table 5.1: Ratios of feasible tasks and traversable edges for each agent type.

Agent Type	Feasible Task Ratio	Traversable Edge Ratio
Adult without motor disabilities	1	1
Wheelchair user	0.48	0.71
Humanoid robot	0.22	0.43
Quadrupedal robot	0.97	0.96
Sweeping robot	0.57	0.79

5.2.2.4 Metrics and Scoring

We evaluate model outputs against ground truth using four complementary metrics, reported both overall and per-embodiment. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the space graph, let $P(v_{\text{src}}, v_{\text{tgt}})$ denote a simple path in $\mathcal{G}$, and let $E(\hat{P}) \subseteq \mathcal{E}$ be the ordered edge set of a predicted path $\hat{P}$.

(1) Feasibility Classification (Feas-F1). We report the F_1 score for binary feasibility prediction.

(2) Path Validity (PV). A predicted route $\hat{P} = [v_0, \dots, v_m]$ is considered valid if (i) $v_i \in \mathcal{V}$ for all i , (ii) $(v_i, v_{i+1}) \in \mathcal{E}$ for all i , (iii) $v_0 = v_{\text{src}}$ and $v_m = v_{\text{tgt}}$, and (iv) $\hat{P}$ is a simple path (no node repetitions). We report:

$$\text{PV} = \mathbb{E}[\mathbb{I}(\hat{P} \in \mathcal{P}_{\text{simple}}(v_{\text{src}}, v_{\text{tgt}}))].$$

(3) Route Traversability Accuracy (RTA). Conditioned on $\hat{y} = 1$ and a valid $\hat{P}$, we evaluate the fraction of edges in $\hat{P}$ that are actually traversable under embodiment $\mathbf{a}$:

$$\text{RTA}(\hat{P}, \mathbf{a}) = \frac{\sum_{e \in E(\hat{P})} g_e^{(\mathbf{a})}}{|E(\hat{P})|},$$

where $g_e^{(\mathbf{a})} \in \{0, 1\}$ is the edge-level traversability label. All edges are assigned unit weight so each edge contributes equally. $\text{RTA} = 1$ indicates a fully traversable route, while lower values quantify partial failures. This metric serves a role analogous to path-quality measures such as SPL [Anderson and et al., 2018b], but conditions on embodiment-specific traversability.

(4) Reasoning Validity (RV). For infeasible predictions ($\hat{y} = 0$), the model must (i) return a proposed route $\hat{P}$ and (ii) provide a short rationale $\hat{\rho}$ explaining the failure. Let $\mathcal{R}^*(\hat{P}, \mathbf{a})$ be the set of annotated failure reasons. We define an LLM-as-judge function J_{LLM} that returns 1 if the model’s rationale semantically matches the annotation and 0 otherwise. We assessed this function by manually validating 300 randomly sampled VLM rationales. Human and LLM judgments agreed in 89% of cases, supporting the use of this method for the benchmark. We report RV as the mean of these binary judgments:

$$\text{RV} = \mathbb{E} \left[J_{\text{LLM}}(\hat{\rho}, \mathcal{R}^*(\hat{P}, \mathbf{a})) \right].$$

Composite CapNav Score. We report a composite score that aggregates the above aspects:

$$\text{CapNav} = \lambda_c F_1 + \lambda_p \text{PV} + \lambda_t \overline{\text{RTA}} + \lambda_r \overline{\text{RV}}, \sum \lambda = 1,$$

where $\overline{\text{RTA}}$ averages over positive predictions and $\overline{\text{RV}}$ averages over negative predictions. We report per-embodiment scores and a macro-average across embodiments. By default, we set each λ to 0.25.

Reference Upper and Lower Bounds. We establish lower and upper reference bounds under the proposed evaluation metrics. We simulate a random-walk policy on the navigation graph that ignores scene and agent constraints. This naive baseline achieves a lower-bound score of 29.35%. As an approximate upper reference, we evaluate four human participants, each of whom completes 20 randomly sampled CapNav

tasks. The human-reference study was reviewed and approved by the University of Washington Institutional Review Board (UW IRB; STUDY00024804). Participant CapNav scores range from 48.97% to 74.77% (mean = 60.59%).

5.2.3 Experiments

Our experiments address three research questions:

1. **RQ1:** How do SOTA VLMs perform on CAPNAV?
2. **RQ2:** How does performance vary across models, agent types, and inference settings?
3. **RQ3:** What failure modes reveal model limitations?

To answer these questions, we evaluate 13 modern VLMs on CAPNAV. We report quantitative results stratified by model family, agent type, inference setting, and challenge type. We further analyze common failure cases and consolidate them into a qualitative error taxonomy.

5.2.3.1 Settings

We evaluated 13 VLMs in November 2025, spanning proprietary and open-source models with strong multimodal capabilities, including the Gemini 2.5 family, GPT-4.1, GPT-5 Pro, Doubao-Seed, and Qwen3-VL. For models with a thinking option, we tested both *thinking* and *nonthinking* configurations. We also included Spatial-MLLM [Wu et al., 2025] and Video-R1 [Feng et al., 2025], which explicitly target spatial reasoning. We accessed proprietary models through APIs and deployed open-source models on either one NVIDIA A100 GPU with 80 GB of memory or one A40 GPU with 48 GB of memory.

At inference time, each model received CAPNAV tasks as a triple $\langle \mathcal{S}, \tau, \mathbf{a} \rangle$ (Section 5.2.2.1). A practical limitation of current VLM interfaces is the maximum number of visual tokens or frames accepted in a request. To study this constraint, we evaluated up to four input settings for each model, subject to its interface limits: **16**, **32**, and **64** total frames per clip, and a **1 FPS** setting with a mean of 163 frames per video.

5.2.3.2 Performance

We compare model outputs with the ground-truth navigation graph to compute overall and per-agent performance. Table 5.2 summarizes performance across all models under both *thinking* and *nonthinking* settings.

When a model was evaluated with multiple frame settings, the table reports its best value for each metric.

Table 5.2: CapNav performance across 13 VLMs. Each value is the best performance across all frame settings tested for that model. The left block reports task metrics (Feas-F1, PV, RTA, RV, and CapNav). The right block reports composite scores for five agent types: adults without motor disabilities, humanoid robots, quadrupedal robots, sweeping robots, and wheelchair users. Bold indicates the best value in each block. Blue indicates open-source models, green indicates proprietary models, and orange indicates models designed for spatial reasoning. N/A indicates that the model did not produce a valid value for the metric.

Model	Feas-F1	PV	RTA	RV	CapNav	HUMAN	HUMANOID	QUAD	SWEEP	WHEEL
Non-thinking mode										
Intern-VL3.5-8B [Zhu et al., 2025]	73.65	26.53	39.54	27.17	41.72	53.89	35.61	51.92	51.36	49.43
Intern-VL3.5-14B [Zhu et al., 2025]	75.68	35.82	50.21	25.18	46.72	50.43	42.98	47.69	44.66	47.38
Keye-VL-1.5-8B [Yang et al., 2025a]	77.63	30.84	43.96	36.70	47.28	58.64	53.44	54.37	62.17	60.11
MiMo-VL-7B-RL-2508 [Team et al., 2025a]	57.31	43.61	60.08	29.35	47.59	46.61	38.96	45.38	41.46	50.85
Qwen3-VL-8B-Instruct [Team, 2025]	78.92	43.39	53.56	47.89	55.94	54.20	38.18	50.91	57.50	53.21
GPT-4.1 [OpenAI, 2025a]	75.90	49.00	65.51	32.86	55.82	73.49	40.35	49.74	57.70	56.24
Doubao-Seed-1.6 [Seed, 2025]	80.26	60.00	71.92	35.47	61.91	65.18	47.23	59.49	60.65	60.82
Thinking mode										
Spatial-MLLM-4B [Wu et al., 2025]	75.27	5.04	10.16	N/A	30.15	35.87	14.86	29.48	25.78	22.69
Video-R1-7B [Feng et al., 2025]	74.57	25.50	44.66	4.82	37.39	43.62	26.35	41.66	32.17	31.82
Keye-VL-1.5-8B [Yang et al., 2025a]	73.07	41.61	48.54	38.67	50.47	51.05	30.99	47.59	46.52	45.98
GLM-4.1V-9B [Team et al., 2025b]	73.81	40.12	55.43	33.12	50.62	50.90	33.19	46.01	45.47	42.46
Intern-VL3.5-14B [Zhu et al., 2025]	78.48	42.96	57.90	24.97	51.08	50.21	39.93	51.13	46.27	47.67
Intern-VL3.5-8B [Zhu et al., 2025]	79.55	43.74	55.10	36.50	53.72	51.23	35.34	51.61	53.74	50.05
MiMo-VL-7B-RL-2508 [Team et al., 2025a]	65.10	59.62	65.00	35.33	56.26	46.61	38.96	45.38	41.46	50.85
Doubao-Seed-1.6 [Seed, 2025]	76.16	61.94	71.93	38.44	62.12	60.31	46.20	58.05	61.02	63.35
Gemini-2.5-flash [Comanici et al., 2025]	84.16	65.12	68.96	38.04	64.07	79.95	40.21	59.15	63.07	59.46
GPT-5-pro [OpenAI, 2025b]	86.87	67.90	75.89	34.81	66.37	82.85	46.95	61.74	70.58	64.78
Gemini-2.5-pro [Comanici et al., 2025]	84.30	73.00	79.15	32.29	67.18	85.96	54.47	61.21	70.87	65.51

All evaluated models outperform the random-walk baseline (CapNav = 29.35%). The strongest performance is achieved by proprietary models (notably the Gemini-2.5 family and GPT-5-pro), which exceed the human average (CapNav = 60.59%) but remain below the best human performance (CapNav = 74.77%). Overall, results show a clear separation between top-tier models and the remainder.

Models explicitly designed for spatial reasoning (Spatial-MLLM and Video-R1-7B) perform substantially worse across all metrics, despite claims of strong spatial performance. This indicates that their spatial reasoning training schemes and architectural modifications are not yet sufficient to address the challenges posed by CAPNAV.

We also test the sensitivity of rankings to the CapNav score composition. In addition to equal weight-

ing, we evaluate four alternative schemes that assign weight 0.5 to one component and 1/6 to each of the remaining components. Kendall’s τ between rankings across these schemes averages 0.909, suggesting that conclusions are largely robust to moderate weight changes.

5.2.3.3 Failure-Mode Taxonomy

To characterize systematic errors, we used Gemini 3 to identify recurring themes in 1,000 randomly sampled inference outputs and manually refined these themes into a four-category taxonomy. We then inspected VLM predictions for 1,500 sampled question-answer pairs to estimate the prevalence and distribution of each error type.

Path hallucination: ($N=659$) The predicted path links nonadjacent nodes or violates the specified start or end, indicating incorrect reasoning about scene connectivity.

Obstacle hallucination: ($N=418$) The model reports nonexistent blocking obstacles, such as closed doors or clutter, suggesting an unreliable interpretation of visual evidence.

Dimension neglect: ($N=191$) The model overlooks geometric constraints that block the agent, such as narrow doors, tight clearances, or cluttered passages, reflecting weak grounding between spatial extent and traversability.

Ability hallucination: ($N=10$) The model’s reasoning contradicts the provided agent profile. We observed this failure in smaller open-source models, such as MiMo-VL 7B.

5.2.3.4 Physical Obstacle Types

We observe four major types of physical obstacles in the ground truth: **stairs** ($N=520$), which block humanoid robots, sweeping robots, and wheelchair users; **door sills or floor-height differences** ($N=82$), which block wheelchair users and sweeping robots; **narrow pathways** ($N=438$), usually caused by narrow doors or furniture clutter, which can restrict humanoid robots and wheelchair users; and **insufficient turning space** ($N=28$), which restricts wheelchair users and, in rare cases, quadrupedal robots. For navigation tasks labeled nontraversable because of one or more of these obstacles, we evaluate how often VLMs both predict infeasibility correctly and provide an appropriate reason. Figure 5.5 shows the results.

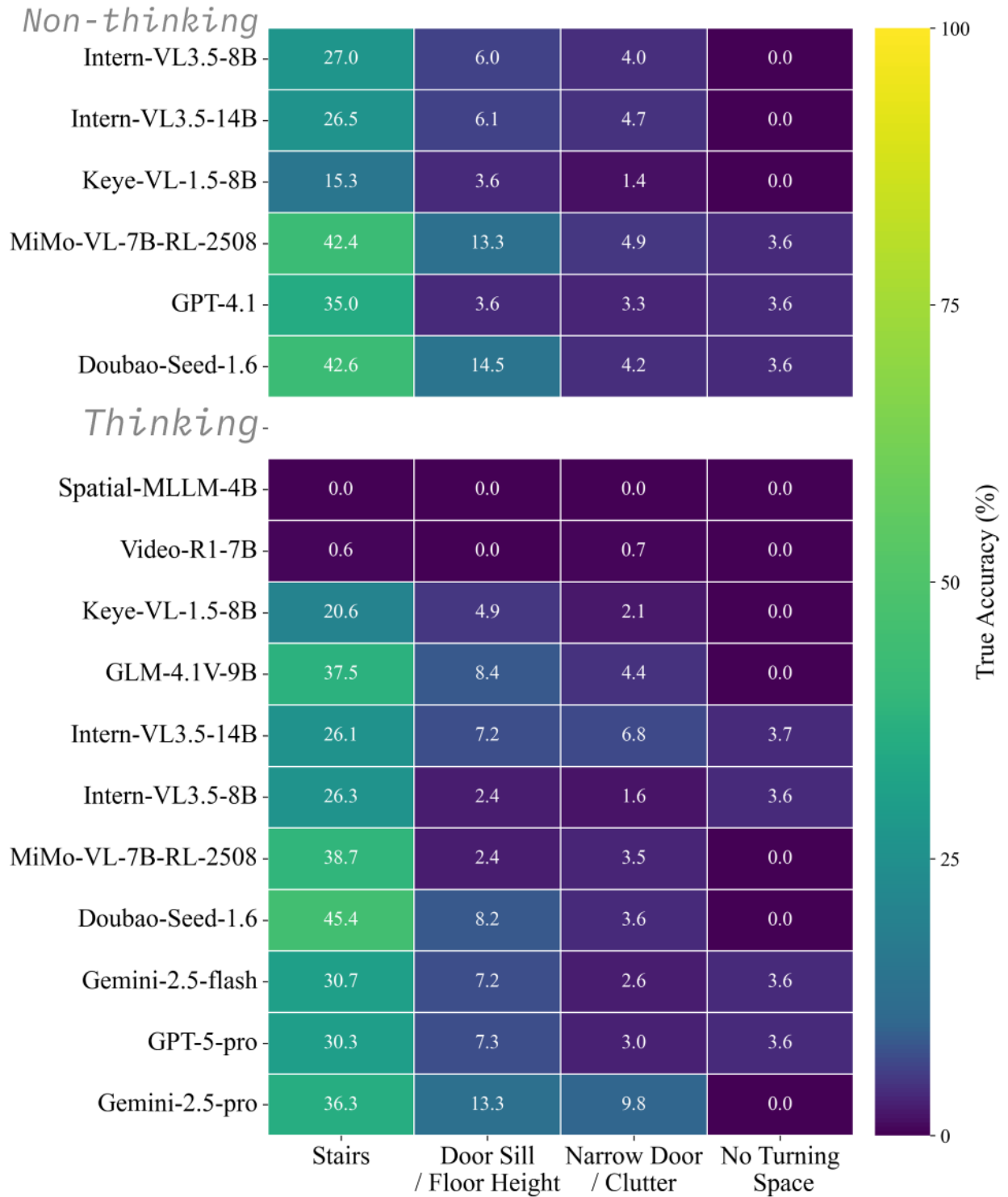


Figure 5.5: Model accuracy by obstacle type.

5.2.4 Findings

Drawing on both quantitative results and qualitative inspection of model outputs, we distill three primary findings. Collectively, they indicate that current VLM navigation remains brittle under capability constraints, visually demanding inputs, and geometry-sensitive traversability judgments.

5.2.4.1 Capability Constraints Induce Systematic Performance Degradation

Across all tested models, performance is highest under the baseline embodiment without mobility limitations (adults without mobility constraints; mean CAPNAV score = 57.83%). As constraints tighten, performance degrades. The largest gap appears for the humanoid embodiment due to (i) its inability to climb stairs and (ii) a minimum pathway clearance requirement of 0.9 m; this configuration attains the lowest average score (mean = 39.12%).

This pattern highlights systematic deficits and imbalances in VLM-based navigation under mobility constraints. Strong performance in baseline settings does not translate to capability-constrained scenarios; hence, targeted evaluation is essential before deployment in safety-critical navigation.

The five agent types supported in CAPNAV provide controlled examples of human and robotic mobility profiles. Practitioners can use the closest profile as an initial reference, but should not assume that it captures an individual person or a substantially different robot. To support evaluation beyond these profiles, we release our annotation interface and data-curation pipeline so that researchers can define new agent profiles. These resources facilitate capability-aware evaluation for additional hardware platforms and mobility conditions.

5.2.4.2 Increasing the Visual Budget Helps but Exposes a Bottleneck

Across nine paired inference experiments with and without thinking mode, enabling thinking mode improves the CAPNAV score by 6.87 percentage points on average. This gain carries a substantial computational cost: mean inference time increases more than eightfold, from 14.94 s to 123.94 s.

Across frame settings, latency remains stable, but higher frame counts increase GPU-memory and API costs. Performance improves unevenly with more frames, tending to improve more for stronger models, notably Gemini, and negligibly for weaker models (Figure 5.6). This pattern reveals a visual bottleneck:

additional evidence helps only when the model can reliably integrate it.

Qualitative inspection suggests two explanations. First, a 64-frame input is already computationally demanding for most open-source models, yet remains sparse relative to a large scene and requires cross-frame reasoning that weaker models lack. Second, additional viewpoints can increase *obstacle hallucinations*: models may interpret salient but nonblocking cues as traversal barriers, echoing prior observations about limitations in VLMs’ nonlocal reasoning [Berman and Deng, 2025].

These results imply a cost-benefit frontier rather than a monotonic more-is-better rule. For CAPNAV-like VLN tasks, increasing the inference budget can improve performance, but the return depends on model capacity and may trade off against latency, cost, and error modes.

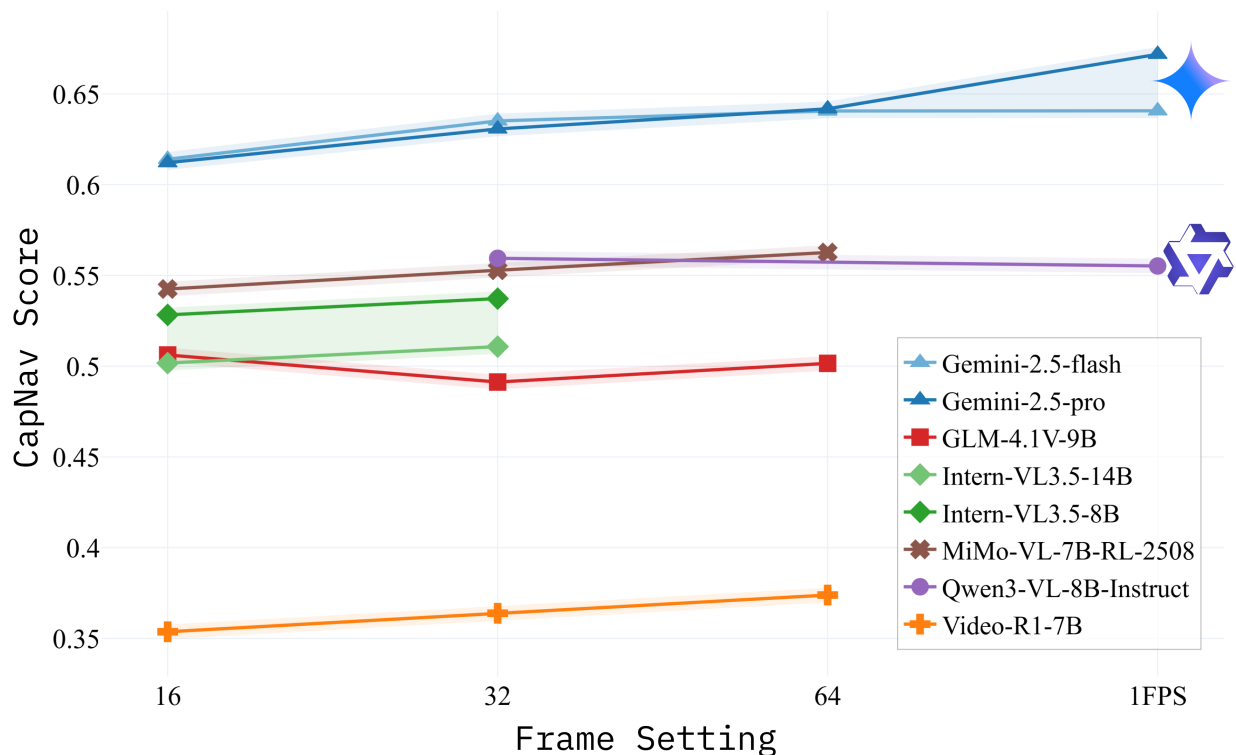


Figure 5.6: CapNav performance across visual-frame-budget settings.

5.2.4.3 Dimension Neglect

In the obstacle-type analysis shown in Figure 5.5, all tested models perform markedly better on obstacles with salient visual signatures, such as stairs and door sills, than on constraints that require implicit metric estimation, such as narrow clearances and turning radii. These failures indicate a recurring form of *dimen-*

sion neglect: traversability decisions that depend on estimating size, clearance, or maneuvering space from images remain challenging for all tested VLMs.

To probe whether these deficits are learnable, we ran a pilot fine-tuning experiment. We augmented the CapNav dataset using the navigation graph to expand the task count from 2,000 to 13,000, then fine-tuned Qwen3-VL-8B-Instruct with LoRA. After one training epoch, the test score improved from 45.26% to 55.18%, and the fine-tuned model produced more dimension-aware reasoning. However, reasoning validity decreased ($0.30 \rightarrow 0.25$), and we observed frequent false positives on narrow passages, incorrectly flagging feasible routes as blocked.

These outcomes suggest that fine-tuning can reduce dimension neglect but does not fully resolve geometric imprecision and may shift the error profile toward overly aggressive obstacle judgments. Future methods must jointly improve obstacle sensitivity, metric reasoning, and calibration. We plan to explore reinforcement-learning-based tuning with task-level rewards under embodiment constraints and to inject spatial information, such as depth estimates, topological priors, or map-like intermediate representations, to improve metric judgments.

5.2.5 Discussion

We discuss the implications of our findings, outline practical guidelines for applying VLMs to CapNav, and propose directions for future work.

5.2.5.1 Summary of Key Findings

Our evaluation reveals substantial performance disparities across model families, agent embodiments, obstacle types, and input configurations. Proprietary models, particularly Gemini 2.5 Pro, GPT-5 Pro, and Doubao-Seed, form the top tier across nearly all metrics. Open-source VLMs, including models explicitly designed for spatial reasoning, perform substantially worse.

Performance improves with larger model sizes, thinking-mode inference, and larger visual-frame budgets, indicating that both representational capacity and intermediate reasoning can benefit CapNav. However, these gains carry higher computational costs and latency, complicating deployment in robotics systems that must operate in real time or with limited resources. The magnitude of improvement also varies across

model families, highlighting the need to validate each model rather than assume uniform benefits.

A central finding is the consistent degradation of performance when an agent’s mobility constraints differ from the baseline profile of an adult without motor disabilities. Even state-of-the-art models that perform strongly under the baseline setting struggle with constraints associated with humanoid robots, sweeping robots, and wheelchair users. This result illustrates a critical limitation of applying general-purpose VLMs to specialized embodiments: strong performance in a comparatively unconstrained setting does not transfer reliably to capability-constrained scenarios. Targeted evaluation is therefore essential before deployment in safety-critical applications.

Our obstacle-type analysis further shows that VLMs remain unreliable on tasks that require metric reasoning, such as assessing doorway clearance or turning radii. They struggle to derive fine-grained geometric judgments from visual tokens. Consequently, even the strongest models do not yet provide dependable traversability assessments for agents with specific clearance requirements.

Taken together, these findings call for better metric reasoning, more effective use of long-horizon visual inputs, and architectures that robustly integrate geometric scene information. Current multimodal models do not yet support these capabilities reliably.

5.2.5.2 Extending Evaluation to New Capability Profiles

The five agent types in CAPNAV are controlled abstractions of selected human and robotic mobility profiles. Practitioners may use the closest profile as an initial point of comparison, but performance on that profile does not establish performance for a person or robot with different capabilities.

For embodiments that differ substantially, such as robots with unusual locomotion modes or devices with atypical clearance or actuation constraints, CAPNAV’s open-source annotation interface and data-curation pipeline allow researchers to define new profiles and extend the benchmark. This process supports capability-aware evaluation for additional hardware platforms and mobility conditions.

5.2.5.3 Further Use of the CAPNAV Dataset

Beyond benchmarking, CAPNAV provides a valuable resource for improving VLMs’ spatial reasoning. Its combination of video tours, structured traversability graphs, and capability-conditioned QA pairs enables

several research directions:

- Fine-tuning or instruction-tuning VLMs for embodied navigation and spatial understanding.
- Augmenting pretraining corpora with explicit geometric and capability-aware supervision.
- Programmatically generating new navigation QA tasks using the graph structure to increase training coverage.
- Developing hybrid reasoning pipelines that combine VLM perception with geometric or classical planning modules.

These avenues can help narrow the gap toward capability-aware navigation and support VLMs that reason more reliably about embodiment-specific mobility constraints in real-world spaces.

5.2.6 Conclusion

We present CAPNAV, a benchmark that systematically evaluates capability-conditioned navigation across human and robotic embodiments, realistic mobility obstacles, and graph-structured indoor environments. Our evaluation of 13 state-of-the-art VLMs reveals three consistent patterns. First, performance degrades as mobility constraints increase, indicating that strong results for the baseline adult profile do not transfer reliably to other embodiments. Second, increasing the inference budget through additional frames or thinking mode yields inconsistent gains and exposes a visual-integration bottleneck. Third, models exhibit dimension neglect when navigation depends on narrow clearances, turning radii, or other metric constraints. A pilot fine-tuning study suggests that these deficits are partially learnable, but can introduce false-positive obstacle judgments. We release the CAPNAV dataset and annotation tool to support more reliable capability-conditioned navigation research.

5.3 Frontier Synthesis

CapNav shows that spatial alignment cannot be inferred from generic navigation performance. The same environment and destination can yield different feasible routes for different embodiments, yet models performance degrade when a task requires them to integrate capability constraints with topology, metric clearance,

turning space, and evidence distributed across views. Errors also occur at distinct levels: a model may misclassify feasibility, propose an invalid path, overlook an untraversable spatial detail, or give an explanation that does not match the physical obstruction.

The benchmark makes this alignment problem inspectable. Capability profiles state assumptions about an agent; navigation graphs expose route structure; edge annotations identify where a capability and environment conflict; and decomposed metrics prevent a plausible final answer from concealing an invalid path or capability violation. At the same time, the five profiles remain controlled abstractions rather than complete descriptions of lived mobility or robotic operation. CapNav therefore supports diagnosis of model sensitivity to specified constraints, not a claim that any one profile fully represents the people or machines named by it.

For human-AI symbiotic spatial perception, the central lesson is that a spatial judgment must answer not only *what is present?* or *where is the goal?*, but also *for whom is this action possible, under which assumptions, and on what evidence?* Frontier III contributes an evaluation scaffold for making those questions explicit before a model is entrusted with consequential guidance.

Chapter 6

Frontier IV: Human-Centered Spatial Representation and Interaction

6.1 Preamble

The fourth frontier concerns the representational interface between human and machine spatial understanding. People form spatial knowledge through embodied movement, visual depth cues, sound, objects, surfaces, landmarks, and mediated forms such as maps and video. AI systems, in contrast, operate through language, images, videos, maps, point clouds, meshes, scene graphs, coordinates, learned features, and generated media. These representations overlap, but they differ in what they make easy to express, inspect, compare, and change. Human-AI symbiotic spatial perception requires a shared representation that is both computationally useful and intuitively perceivable and actionable.

Language has become a convenient bridge in human-AI collaboration, but it is an incomplete one for spatial work. Text prompts can state a broad range of goals, yet they often underspecify nuanced intent concerning scale, position, timing, orientation, depth, and object relationships. As a result, users of generative AI tools can become trapped in iterative prompt refinement: each new prompt repairs one inferred detail while introducing another ambiguity.

Recent HCI research points toward richer alternatives. A recent review documents the growing use of direct-manipulation methods alongside language in generative image tools [Park et al., 2026]. InterChat

shows how direct manipulation of visual elements can complement natural-language instructions when users need to communicate complex visual-analytic intent [Chen et al., 2025]. In spatial and mixed-reality authoring, Reality Proxy, Brickify, and ImagineAR similarly use manipulable objects, design tokens, or in-situ interaction to make intent more concrete and revisable [Liu et al., 2025; Shi et al., 2025; Lee et al., 2025]. These systems do not eliminate language; rather, they give language an appropriate partner when the intended relation can be pointed to, heard, positioned, or directly adjusted.

Creative spatial tasks make the need for such a partner especially clear. A creator may know that a sound should fit a particular physical interaction, or that a design element should sit behind a subject and follow the geometry of a scene, without being able to specify every relation in prose. Conventional authoring tools can offer precision, but often by exposing implementation details and imposing a steep learning curve. Generative AI can reduce this technical burden only if it also exposes meaningful handles through which people can express preferences, assess the inferred structure, and retain control over the result.

The projects in this chapter investigate two complementary forms of this shared surface. SonifyAR considers how sensed spatial context can be translated into an auditory output that people assess in situ. DepthScape considers how AI-inferred depth and semantics can become direct-manipulation handles for visual authoring. Together, they develop the final frontier of the thesis: representations that make AI spatial inference perceptible, expressive, and open to human revision.

6.2 SonifyAR: Context-Aware Sound Generation in Augmented Reality

SonifyAR investigates sound as a human-centered representation of spatial events in augmented reality. An AR sound depends not only on an action label such as tapping or sliding, but also on who or what caused the event, which virtual object participated, which physical surface or scene received the action, and how those entities should sound together. SonifyAR formalizes this context as a triad of user, virtuality, and reality, textualizes event relationships, and uses an AI controller to recommend, retrieve, generate, or transform sound assets.

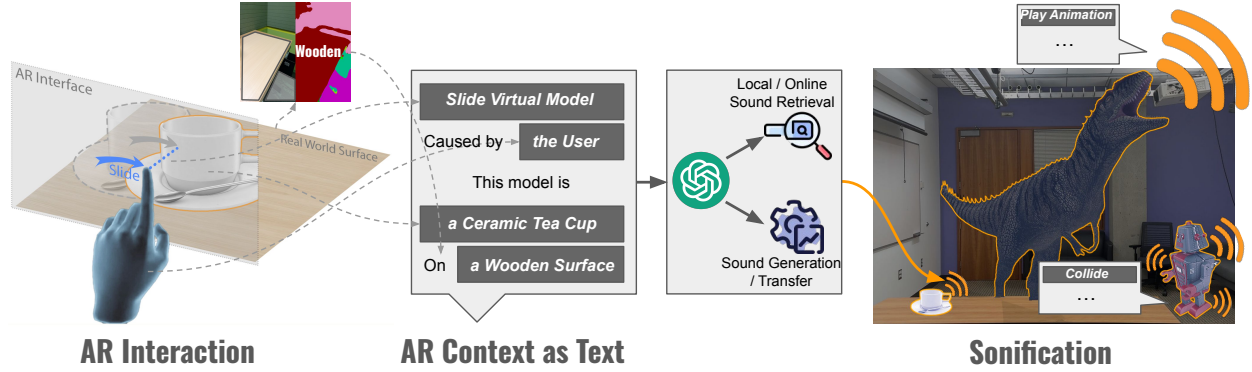


Figure 6.1: SonifyAR observes an AR event and its user, virtual-object, and physical-scene context, then recommends, retrieves, generates, or transforms a matching sound that can be tested in situ.

6.2.1 Introduction

Sound is a critical but often overlooked element in augmented reality (AR). AR sound can improve immersion and overall user experience [Roginska and Geluso, 2017]; support depth perception, task completion [Zhou et al., 2007], search, and navigation [Rumiński, 2015]; and assist people with low vision [Ribeiro et al., 2012b]. Despite its importance, current AR authoring platforms such as Reality Composer [Apple, Accessed September 11, 2023], Adobe Aero [Adobe, Accessed September 11, 2023], and Unity Mars [Unity Technologies, 2023] provide only rudimentary sound support. We identified three critical gaps in existing AR authoring tools:

1. **Limited real-world context.** Existing systems typically support action triggers linked to virtual objects but lack real-world contextual information, such as the surface traversed by a virtual toy robot.
2. **Limited interaction specification.** Existing systems provide only predefined interaction triggers such as “Tap” and “Proximity Enter.” This limits a creator’s ability to specify interactions outside the provided options, particularly those involving environmental context, such as sliding virtual chalk on a real-world blackboard.
3. **Limited sound sources.** Existing systems are limited by the sound assets in their libraries and the scarcity of suitable online resources. As a result, AR authors struggle to find sounds for distinct AR events, such as a virtual dragonfly’s wing flutter or a virtual dinosaur eating.

To address these challenges, we present SonifyAR, a context-aware AR sound-authoring system that

uses GPT-4 [OpenAI, 2024] and the text-to-audio diffusion model AudioLDM [Liu et al., 2023]. SonifyAR makes three key technical contributions:

First, SonifyAR uses programming by demonstration (PbD) [Lau and Weld, 1998] to simplify the specification of complex AR interactions. The PbD pipeline lets users demonstrate interactions while the system detects the action and collects its context. For example, a creator who wants to sonify a walking robot can place it on a physical surface. As the robot walks, SonifyAR captures the collisions between its feet and the surface, together with information such as the robot’s attributes and the surface material.

Second, SonifyAR represents context as text. Text provides a common representation for AR information drawn from multiple sources, including user actions, virtual objects, and the physical environment, and from multiple formats, including categorical data, 3D shapes, and images. The system generates an LLM prompt with the following template: “This event is [Event Type], caused by [Source], and acts on [Target Object]. [Additional Information on Involved Entities].”

Finally, SonifyAR supports LLM-based sound acquisition. It integrates four methods, recommendation, retrieval, generation, and transfer, under the control of an LLM. For each AR interaction, the LLM receives the textual context and returns commands that invoke these methods to provide matching sound assets.

To author AR sound with SonifyAR, a creator initiates AR interactions and the system automatically lists context-matched sound options. For example, a creator may want a virtual teacup to chime appropriately while sliding across a table (Figure 6.1). In a traditional workflow, the creator would manually specify the sliding action and locate a sound asset for the ceramic cup. With SonifyAR, the creator demonstrates the action and then chooses from automatically generated sound options.

To explore SonifyAR’s potential, we conducted a qualitative user study with eight designers and developed five application scenarios ranging from AR education to headset safety. The user study examined the prototype’s usability and participants’ reactions, identifying both perceived benefits and opportunities to improve sound quality and interface design. The application scenarios illustrate how automatically acquired sounds could be incorporated into varied AR experiences that would otherwise require substantial manual effort.

Our work makes three primary contributions: (1) a design space for AR sound-authoring tools that highlights gaps in existing systems; (2) SonifyAR, a context-aware sound-authoring pipeline that incorpo-

rates sensed environmental cues such as surface material, virtual-object semantics, and user actions; and (3) findings from a user study with eight designers and five application examples that characterize SonifyAR’s potential and current limitations. To our knowledge, SonifyAR is the first system to offer automatic, in-context sound authoring for AR.

6.2.2 The Design of SonifyAR

Three research questions guided the design of a sound-authoring system that generates sounds matched to AR interactions:

- What are the target AR interactions that should produce sound effects?
- How can we efficiently acquire sound assets for these AR interactions?
- How can these AR interactions be specified and represented in the authoring process?

This section develops each question and the corresponding design rationale.

6.2.2.1 The Triad of User, Virtuality, and Reality

We began by examining the design space of sounds for AR interactions. Drawing on Jain et al.’s [Jain et al., 2021] sound taxonomy for VR, we conducted a similar analysis for AR. We examined several widely used code-free AR authoring tools, including Adobe Aero [Adobe, Accessed September 11, 2023], Apple Reality Composer [Apple, Accessed September 11, 2023], and Halo AR [hal]. These tools support sounds related to virtual-content interactions or spatial anchors, such as playing a sound when a user approaches a location, scans a QR code, or taps a virtual model.

The literature on AR sound [Dam et al., 2024; Rakkolainen et al., 2021; Filus and Rambli, 2012], however, envisions a much broader scope. Rakkolainen et al. [Rakkolainen et al., 2021] classified AR audio into five classes: ambient, directional, musical, speech, and noise. Dam et al.’s [Dam et al., 2024] taxonomy of audio augmented reality (AAR) classifies AR sound as environment-connected, goal-directed, and context-adapted, as well as at the intersections of these categories. This taxonomy emphasizes that AR sound is often rooted in physical context and user intention.

Drawing on this prior work, we introduce an AR sound design space that emphasizes the sonic interplay among three elements of an AR experience: *reality*, the physical environment that hosts the experience; *virtuality*, the virtual objects placed in the experience; and *the user*, the person interacting with it. Below, we describe each element and indicate its level of support in existing code-free AR sound-authoring tools [Adobe, Accessed September 11, 2023; Apple, Accessed September 11, 2023; hal].

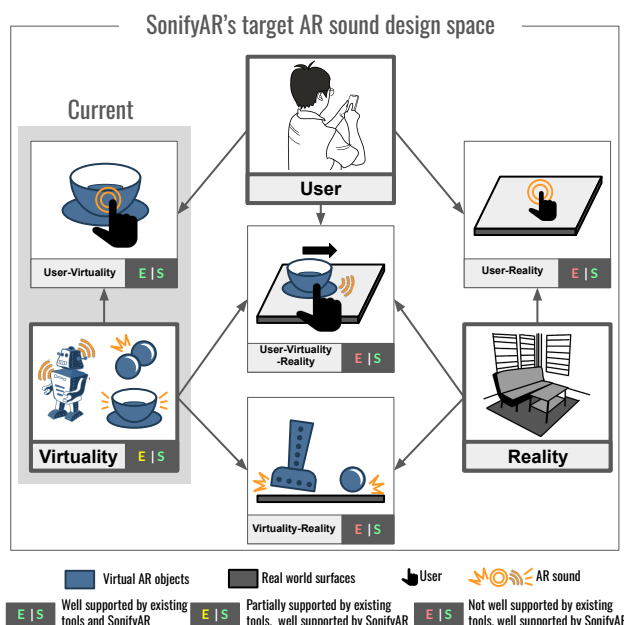


Figure 6.2: Sound-producing opportunities in the triad of User, Virtuality, and Reality.

Virtuality (*partially supported*). *Virtuality* refers to sounds that accompany AR events involving only virtual objects. A sound could correspond to an object’s change of state (e.g., a notification when a virtual object appears), an animated behavior (e.g., the mechanical noise of a virtual dinosaur roaring), or an interaction between virtual objects (e.g., two virtual balls clacking when they collide). Current authoring tools support state changes and animations as sound triggers but do not support sounds for interactions between virtual objects.

User-Virtuality (*well supported*). *User-Virtuality* refers to sounds that respond to users’ actions on virtual objects, such as a virtual dog barking when a user taps or approaches it. All the AR creation tools we examined support user-initiated triggers such as “Tap” and “Proximity Enter.”

User-Reality (*not well supported*). *User-Reality* refers to sounds that accompany a user’s interaction with the physical environment through an AR device, such as a material-appropriate sound when the user

taps a physical table through a phone screen. None of the AR authoring tools we examined supports User-Reality actions as triggers.

Virtuality-Reality (*not well supported*). *Virtuality-Reality* denotes sounds produced when virtual objects interact with the physical environment, such as a virtual robot stomping on a glass surface. Although some systems support physics simulations between virtual models and physical surfaces, the authoring tools we examined do not expose these interactions as sound triggers.

User-Virtuality-Reality (*not well supported*). User-Virtuality-Reality refers to sounds that accompany user actions involving both virtual and physical elements, such as material-aware sliding sounds when a user moves a virtual scraper across a wooden table, painted wall, or glass window. None of the authoring tools we examined supports specification of this complex interaction.

Other Categories. We exclude the Reality and User domains because our interest lies in events for which an AR system, such as a smartphone or headset, generates sound. Sounds produced naturally by a user or the physical environment, such as a person knocking on a wooden table, are therefore outside this discussion.

As illustrated in Figure 6.2, the AR authoring tools we examined do not support many parts of this design space, particularly those involving *Reality*. SonifyAR aims to address this gap.

6.2.2.2 Acquisition of AR Sound

The design space also highlights limits in existing sound-acquisition methods. A mechanical sound for a virtual robot can be readily sourced from local or online databases. More specific sounds, such as a virtual steel ball striking a physical concrete wall or a virtual racecar landing in a backyard pool, require physical-world understanding and material awareness. The possible effects are nearly infinite, so text-to-sound generation is useful in these cases. Conversely, highly recognizable sounds, such as a dog bark, remain difficult for current generative models and can yield noisy or unsatisfactory outputs. The diversity of creator needs and the tradeoffs among acquisition methods suggest that a combination of generation and retrieval could provide a practical solution.

6.2.2.3 Event Representation

Combining generation and retrieval raises the question of how to orchestrate these sound-acquisition methods from a common representation. While visual formats such as images [Sheffer and Adi, 2023] and videos [Zhou et al., 2018; Ghose and Prevost, 2022] are widely used as inputs to sound-generation models, they do not fully capture AR context, particularly user actions. We therefore use text as a universal representation for sound acquisition. Text can precisely convey the context and event specifics, for example, “User slides a ceramic cup on a wooden surface.” Because AR experiences run on hardware-software systems with multimodal sensors, including cameras, IMUs, and GPS, the system can monitor and summarize events as descriptive text. Recent work such as HuggingGPT [Shen et al., 2023], which uses an LLM to control AI tasks, also demonstrates the practicality of LLMs as controllers for sound generation.

6.2.3 SonifyAR Implementation

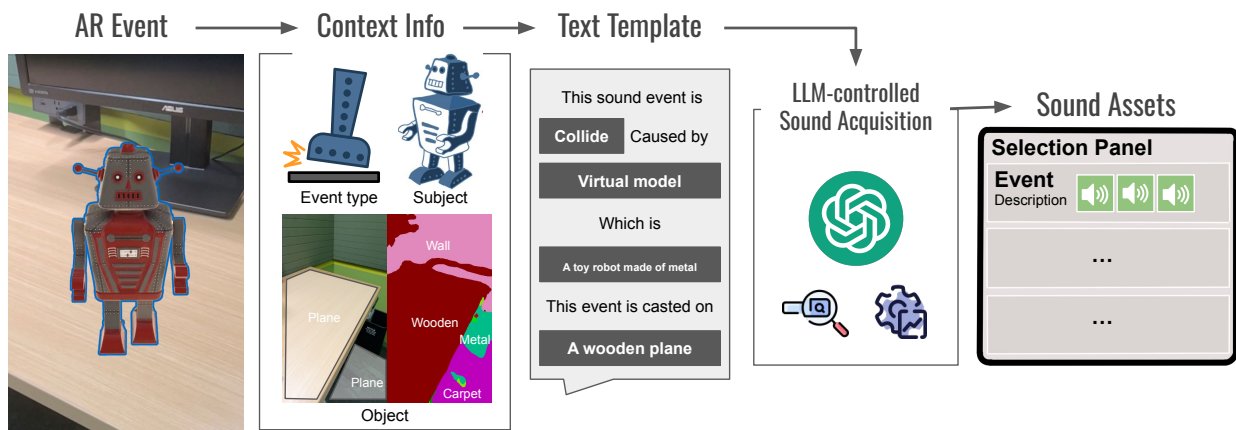


Figure 6.3: Overview of the SonifyAR pipeline. The system monitors and logs AR-event context, including the event type, participating virtual and real-world entities, and their attributes such as material. It compiles this information into a text template, which an LLM controller processes to acquire sound assets. The resulting options are presented in a selection panel.

Building on the design space above, we introduce SonifyAR, a PbD sound-authoring framework that uses context recognition and generative AI to create context-sensitive sounds for AR interactions (Figure 6.3). SonifyAR has three primary components: (1) event textualization, which recognizes a user action as an AR event and transforms its context into a textual description; (2) sound acquisition, an LLM-controlled process that uses four methods to produce context-matched sound effects; and (3) a user interface

through which creators experience the AR scene and view, modify, and test sounds associated with detected events.

6.2.3.1 Event Textualization

In SonifyAR, we use a PbD authoring framework to capture potential sound-producing AR events and compile their context into text. When users interact with the AR space, SonifyAR captures AR events that can produce sound feedback. We define an AR event as a user action and its subsequent result, such as when a user taps a virtual object to trigger its animation. Each event is represented by an event type, action source, and action target. Information about participating virtual objects and real-world planes is also included in the context. The resulting event data is transformed into text with a template.

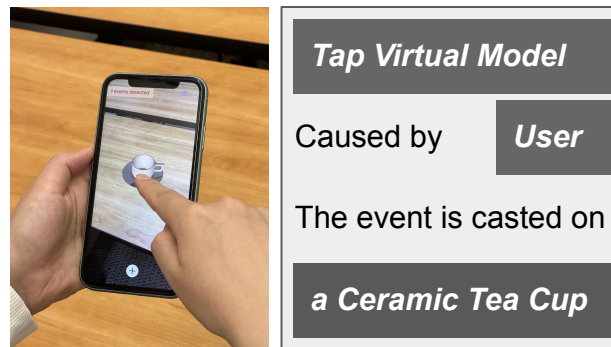


Figure 6.4: SonifyAR’s event textualization. Left: a user taps a virtual cup through the SonifyAR interface. Right: SonifyAR’s PbD framework extracts a textual representation of the event context.

Event Types. To support a broad range of AR sound interactions, we implemented six event types:

1. Tap real-world structure: a user taps a real-world structure through the AR interface.
2. Slide: a user holds a virtual object and slides it on a real-world surface.
3. Collide: a virtual object collides with another virtual object or a physical surface. We implement a specialized collision-detection mechanism, described in subsubsection 6.2.3.4.
4. Appear: a virtual object appears in the AR experience.
5. Tap virtual object: a user taps a virtual object through the AR interface.

6. Play animation: a virtual object plays an animation.

Scene Context Understanding. SonifyAR uses ARKit’s plane-detection functionality [Apple, 2023a,b] to assess the surrounding environment. Because many AR experiences are anchored to planes, detected-plane information provides important context. We enrich this information with the Dense Material Segmentation model [Upchurch and Niu, 2022], which segments the scene and identifies plane materials (Figure 6.3). These labels help SonifyAR produce material-specific sound effects when an AR event involves a plane. We currently support six materials: wood, carpet, concrete, paper, metal, and glass. Surfaces outside these categories are labeled `unknown surface`.

Virtual Object Understanding. Virtual-object semantics are also important for sound generation. To align sound assets with an object’s material and state, we collect text descriptions of each virtual object (e.g., “This model is a toy robot made of metal.”) and its animations (e.g., “A toy robot walks.”). The asset creator can provide these descriptions, or SonifyAR can prompt the user to add relevant details. An event involving the object or its animation incorporates the corresponding descriptions.

Event-to-Text. To aggregate the multi-source context described above, we use a simple template for the LLM-based backend: “This event is [Event Type], caused by [Source], and acts on [Target Object]. [Additional Information on Involved Entities].” *Event Type* identifies the event; *Source* identifies its subject, which may be the user or a virtual object; and *Target Object* identifies the object acted on, such as a tapped plane or a played animation. *Additional Information on Involved Entities* contains details that further describe the source, event, and target, including material descriptions and animation details. We leave the corresponding entity field blank when events are missing a target object or additional information.

During user interactions, the system monitors and logs events in the AR space. It retrieves the associated context and fills the template to create a description of the interaction.

6.2.3.2 Sound Acquisition

SonifyAR uses GPT-4 [OpenAI, 2024] to retrieve or generate context-matched sound assets for an AR event. Inspired by HuggingGPT [Shen et al., 2023] and Visual ChatGPT [Wu et al., 2023], we use the LLM as a

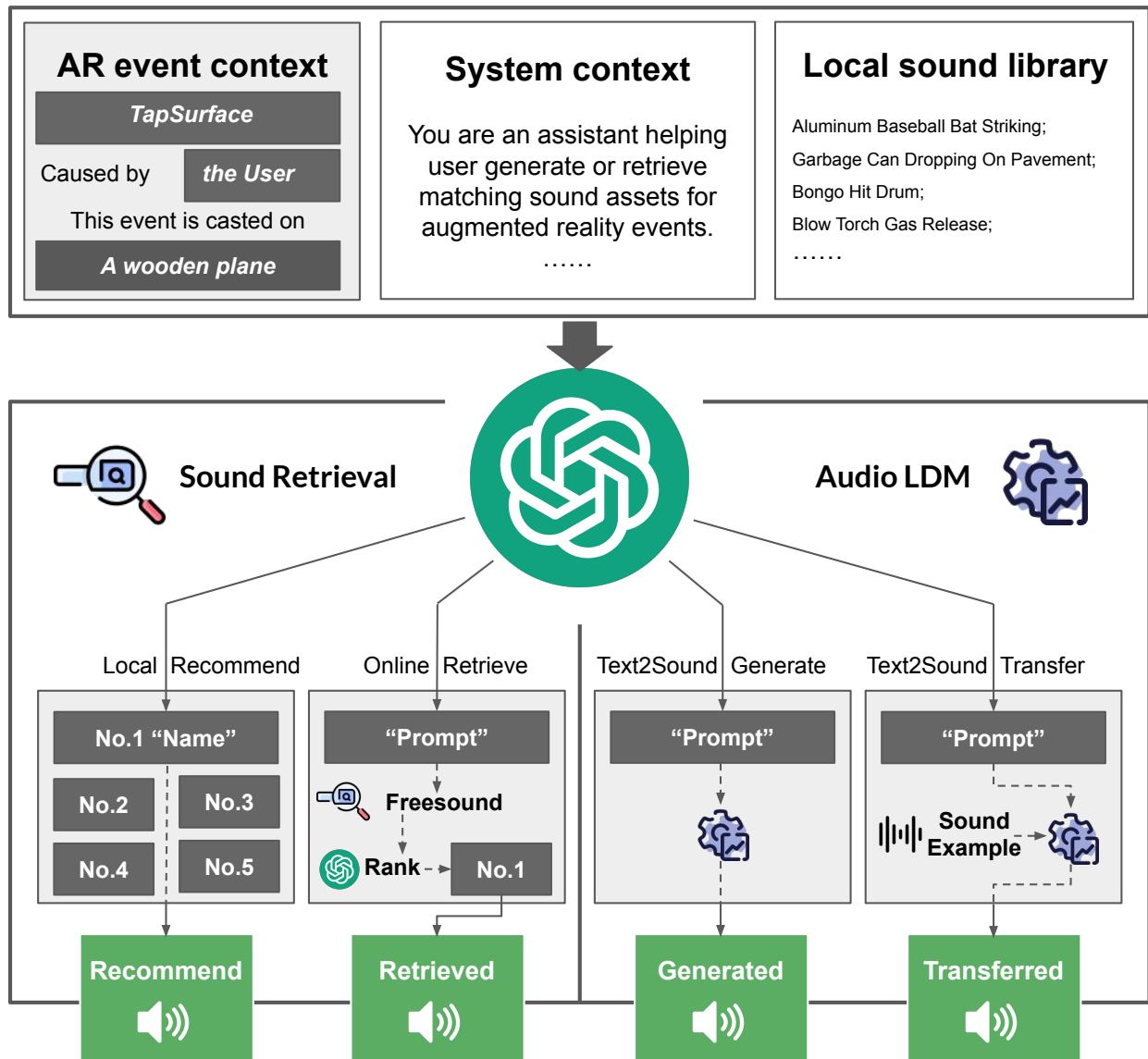


Figure 6.5: SonifyAR’s sound acquisition pipeline.

controller for multiple sound-authoring methods. The LLM takes an event description as input and returns commands for sound acquisition (Figure 6.5). We support four methods: local recommendation, online retrieval, text-to-sound generation, and text-guided sound transfer. These processes operate concurrently in the backend to preserve an uninterrupted AR experience.

Local Recommendation. The LLM can recommend sound assets stored in a local database based on the event description. As in other AR authoring tools, we draw sound effects from the Adobe Audition library [Adobe, 2023]; each has a descriptive filename, such as “Crash Aluminum Tray Bang” or “Liquid Mud Suction.” We provide the complete catalog of filenames to the LLM, which recommends five sounds based on the event context. It returns each selection in the format `method1recommend:FILENAME`, where `FILENAME` is replaced by the actual filename. SonifyAR parses the response and adds the highest-ranked sound to the options for that event. Users can long-press the option to reveal the other recommendations.

Online Retrieval. We extend retrieval with the online sound-asset database Freesound [fre]. Its API returns a ranked list of sound effects for a query. The LLM condenses the event description and returns the query in the format `method2retrieval:PROMPT`, with `PROMPT` replaced by the search terms. The Freesound API returns results as JSON strings. SonifyAR downloads the five highest-ranked matches and presents them as sound options.

Sound Generation. Beyond recommendation and retrieval, we generate custom sound effects with the audio-diffusion model AudioLDM [Liu et al., 2023]. We ask the LLM to compress the event description into a generation prompt in the format `method3generation:PROMPT`, where `PROMPT` is replaced with the generation prompt. Upon receiving this command, SonifyAR sends the prompt to AudioLDM to request text-to-sound generation. The generated sounds are then shown in the UI as options.

Sound-Style Transfer. In addition to text-to-audio generation, AudioLDM can perform text-based sound-style transfer. For events such as tapping, sliding, or colliding, SonifyAR uses a default sound effect rather than generating one from scratch and applies a text prompt provided by the LLM, for example, to transform a generic tap into a tap on glass. This approach preserves the input sound’s length and rhythm, allowing it to remain well timed with the action. SonifyAR also offers text-based style transfer for users who wish to further modify a provided sound asset.

6.2.3.3 User Interface

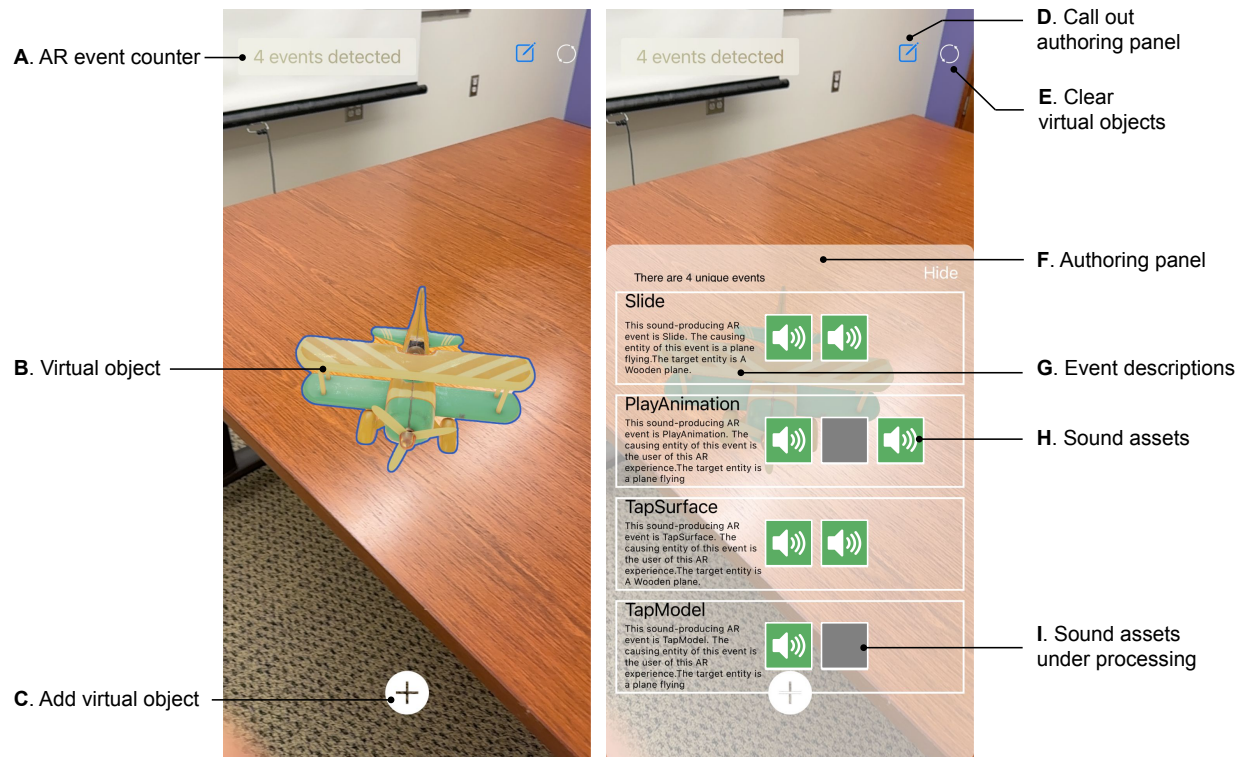


Figure 6.6: SonifyAR’s authoring interface. Left: SonifyAR’s phone-based AR interface. Right: SonifyAR’s authoring panel.

As a PbD authoring framework, SonifyAR invites users to explore an interactive AR experience (Figure 6.6, left). Users can move virtual objects or interact with physical surfaces. When SonifyAR detects an AR event, a text label appears in the upper-left corner of the screen (Figure 6.6A), and the sound-acquisition component generates candidate sounds. Selecting the authoring-panel button (Figure 6.6D) opens an editing interface (Figure 6.6F) over the AR scene. The interface lists detected AR events (Figure 6.6G) and their generated sound assets (Figure 6.6H). Users tap a sound to preview it and double-tap to select and activate it. They can then hide the panel, resume the AR experience, and test the selected sounds. Reopening the editing interface allows them to revise these choices.

Additionally, users can long-press a sound asset to open a menu of exploratory options. For recommended or retrieved sounds, the menu lists the remaining top-five recommendations. For every sound asset, it provides style-transfer and similar-sound-generation options. Users can enter a short text prompt to guide

generation, then iterate on the resulting variations.

6.2.3.4 Technical Implementation

SonifyAR is built with Apple’s ARKit in Swift 5.7 and Xcode 14, and runs as an iOS application on iPhones. Because plane detection performs better on LiDAR-equipped devices, we developed and tested the app on an iPhone 13 Pro Max running iOS 16. We implemented collision detection between virtual objects and planes with ARKit’s physics simulation [Apple, 2023d]. To enable collision detection for animated virtual-object parts, such as a robot foot stomping on a plane, we bind colliders to the joints of the animated bones.

For the LLM-based backend, we use GPT-4 [OpenAI, 2024]. We use the Dense Material Segmentation (DMS) model [Upchurch and Niu, 2022] to obtain detailed material labels and AudioLDM [Liu et al., 2023] for text-to-sound generation and text-guided sound-to-sound transfer. All models are hosted on a server with an NVIDIA RTX 4080 GPU and CUDA 12.2.

6.2.4 User Study

To examine SonifyAR’s usability and support for sound authoring, we conducted a study with eight participants. Each participant used the application in a fixed experimental setting and rated several aspects of the system.

6.2.4.1 Participants

We recruited eight participants, six men and two women between 26 and 33 years old, through snowball sampling at Adobe. Participants had varying levels of AR experience. Six had prior experience with AR, and three had used AR authoring tools.

6.2.4.2 Procedure

The study took place in a small meeting room and had four parts. First, we introduced the study, then collected demographic and background information. Second, we demonstrated SonifyAR through an example AR experience and answered participants’ questions. Third, participants independently explored the application. We provided a virtual walking-robot model that they could place on indoor surfaces and activate by

tapping. For consistency, all participants used this asset and the automatic sound-authoring pipeline until they were satisfied with the result. We recorded the screen and logged user operations throughout this activity. Fourth, participants rated SonifyAR’s usability, helpfulness, and technical performance and explained their ratings. Figure 6.7 presents the questions and results.

6.2.4.3 Results

All participants completed the AR sound-authoring task with SonifyAR. The authoring process took 406 seconds on average ($SD=137$ seconds), and the complete study session took approximately 35 minutes. Participants previewed a mean of 56 sound assets ($SD=10.5$) and selected a mean of 6.4 sounds for their authored experiences ($SD=1.7$).

Participant feedback was largely positive. Participants rated SonifyAR’s helpfulness for AR authoring at a mean of 6.0 out of 7 (Q1, $SD=0.93$) and their willingness to use it in their own practice at 6.3 (Q2, $SD=1.16$). They rated the potential for sounds involving physical surfaces to improve AR immersion at 6.4 (Q3, $SD=1.1$). They also preferred the automated authoring process to their prior manual searches for sound assets (Q5, $M=6.1$, $SD=1.0$). Generated-sound quality received a lower mean rating of 4.75 (Q4, $SD=1.39$), indicating an important area for improvement. Because the pipeline is modular, future versions can incorporate more capable sound-generation models. Figure 6.7 presents the full questions and rating distributions.

Participants also suggested improvements to the interface and its feedback. One participant said, “I really want more usability features that can help communicate these events,” and another said, “I hope there are more associations between the AR events and the sound generation happening in the backend, like some real-time audio hints.”

6.2.5 Applications

To further explore and demonstrate the potential of our approach, we authored five AR scenarios with SonifyAR. See the supplementary video for additional details and an audio-visual demonstration of the resulting SonifyAR sound effects.

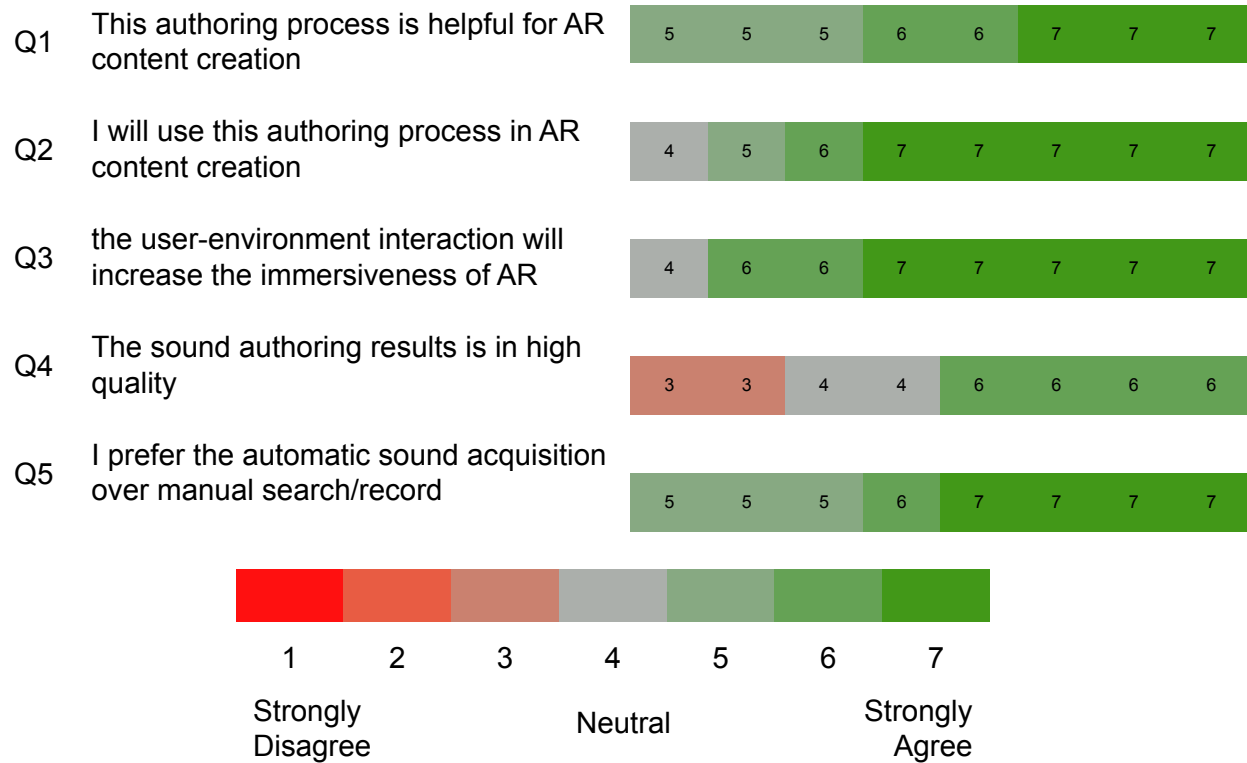


Figure 6.7: Rating questions and response distributions from the SonifyAR user study.

6.2.5.1 Education

AR has been widely explored in STEM education (e.g., [Sirakaya and Alsancak Sirakaya, 2022; Kang et al., 2020, 2019, 2016; Suzuki et al., 2020; Chulpongsatorn et al., 2023]). Prior educational applications simulate and visualize physical or chemical phenomena in AR [Radu and Schneider, 2019; Irwansyah et al., 2018]; SonifyAR could complement these visualizations with context-appropriate sound. To illustrate this possibility, we implemented an AR physics experiment depicting conservation of momentum. The scene contains a downward ramp and two metal balls, one at the top and one at the bottom (Figure 6.8). When the experiment begins, the upper ball rolls down the slope and collides with the lower ball. The collision causes the lower ball to move forward and eventually fall while the upper ball stops. SonifyAR detects the collisions and generates material-specific effects, including the clash of two metal balls and a metal ball striking a table.

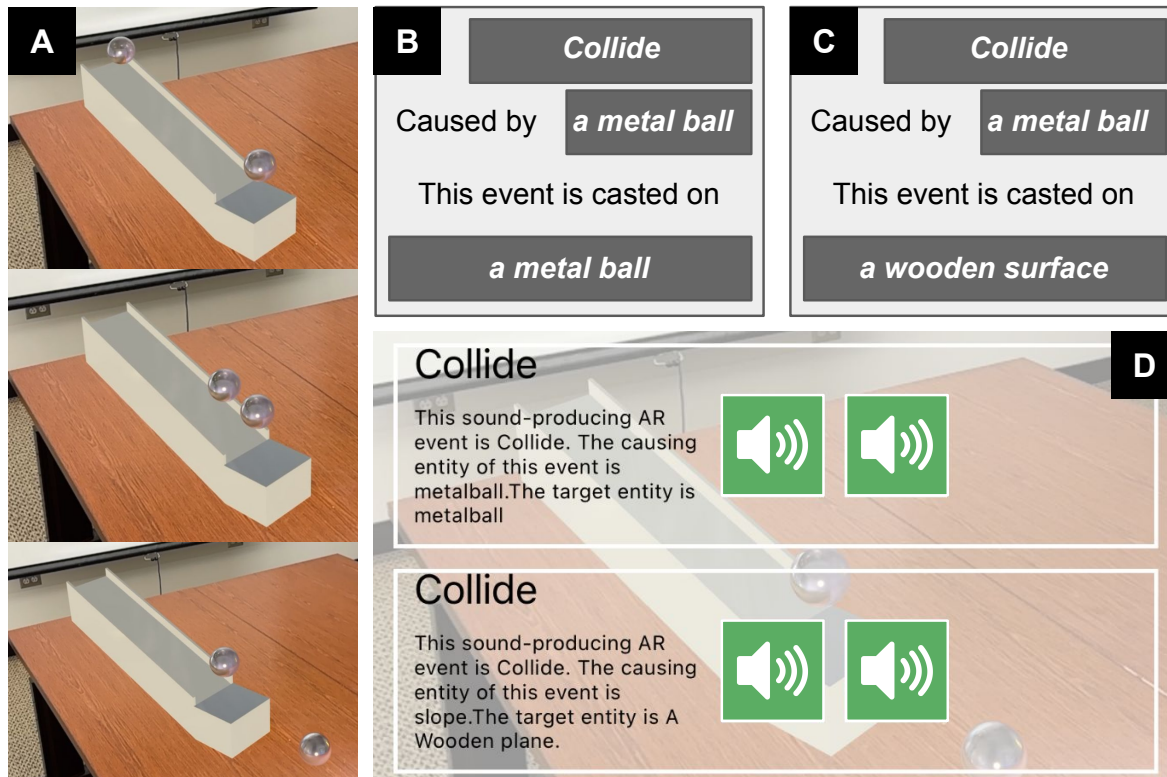


Figure 6.8: An AR physics experiment sonified by SonifyAR. (A) The ramp, metal-ball models, and simulation sequence. (B, C) Contextual information for two collisions in the simulation. (D) Sound assets acquired for the collisions.

6.2.5.2 Accessibility

Audio augmented reality (AAR) has been explored as a form of assistance for people who are blind or have low vision [Ribeiro et al., 2012a; Coughlan and Miele, 2017]. These systems motivate three hypotheses for accessible use of SonifyAR. First, broader support for sound authoring could make it easier for creators to add cues that communicate virtual events. Second, context-aware spatial audio could communicate the position and activity of virtual objects [Ribeiro et al., 2012a]. Third, sounds generated when a user selects a physical surface through AR could communicate information about the surrounding environment. SonifyAR was not evaluated with blind or low-vision participants, so these possibilities require participatory design and direct accessibility evaluation.

To illustrate one of these hypotheses, we use a 3D model of a toy robot (Figure 6.9). As the robot moves virtually from one physical surface to another, such as from wood to carpet, the automatically generated

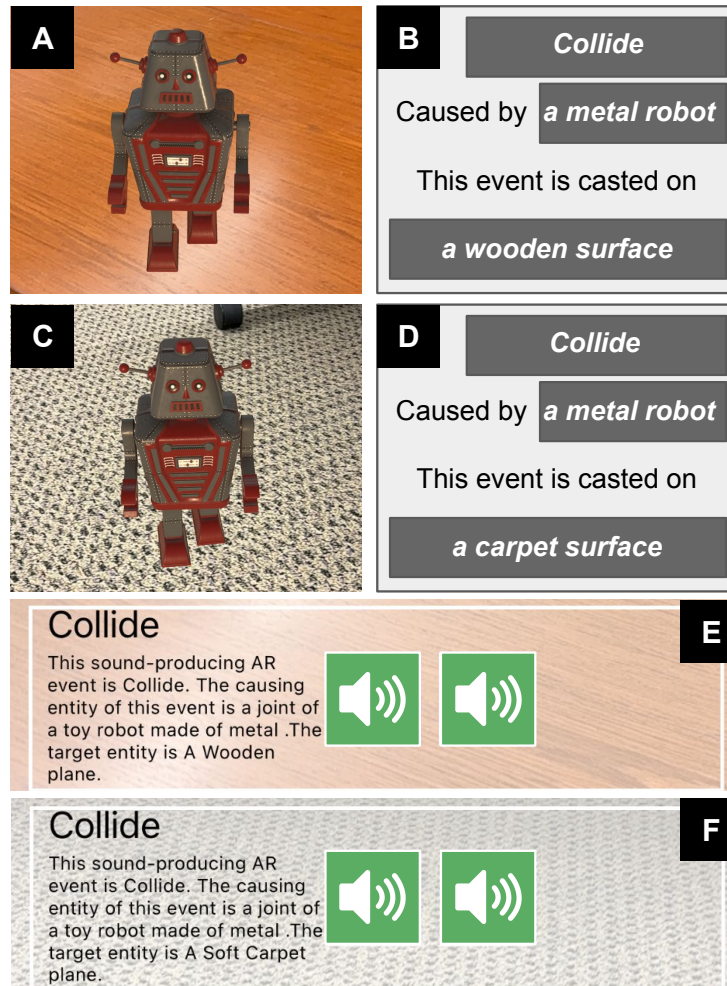


Figure 6.9: A virtual robot walking on different surfaces with different sounds. (A, B) The robot on a wooden surface and the corresponding context. (C, D) The robot on carpet and the corresponding context. (E, F) The acquired sound assets.

sound changes with the detected material. This variation could provide an auditory cue to the robot’s location and activity, but the effectiveness of that cue remains to be evaluated with intended users.

6.2.5.3 Sonifying Existing Applications

Because SonifyAR’s sound-acquisition pipeline accepts textual event descriptions, it could be integrated into existing AR applications. An implementation in an SDK such as ARKit or ARCore could expose descriptions of AR events to the pipeline and return automatically acquired sound options.

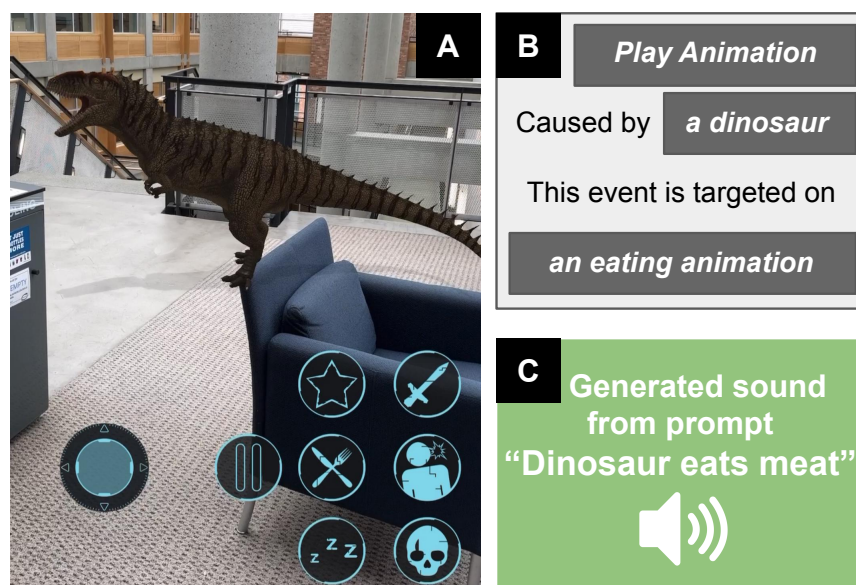


Figure 6.10: An AR dinosaur sonified by SonifyAR. (A) The AR Camera application shows a virtual dinosaur playing an eating animation. (B) Contextual information describes the scene. (C) SonifyAR produces a text prompt and sound output.

We explore this possibility with a Wizard of Oz (WoZ) prototype in the phone-based AR application ARVid [arv], a downloadable app that places animated 3D objects in real-world environments. We selected a dinosaur animation as an example (Figure 6.10) and used WoZ to simulate direct integration of the SonifyAR pipeline into ARVid. By manually supplying context information, for example, “The dinosaur is eating,” to the SonifyAR pipeline, we generated sound assets that matched the 3D animation.

6.2.5.4 Using SonifyAR on an MR Headset

Head-mounted AR and mixed-reality systems place virtual content in close relation to physical surroundings. We use a Wizard of Oz prototype to illustrate how SonifyAR could communicate contact between a virtual window and a physical surface through a material-aware sound. Whether such feedback improves safety or spatial awareness requires a dedicated evaluation.

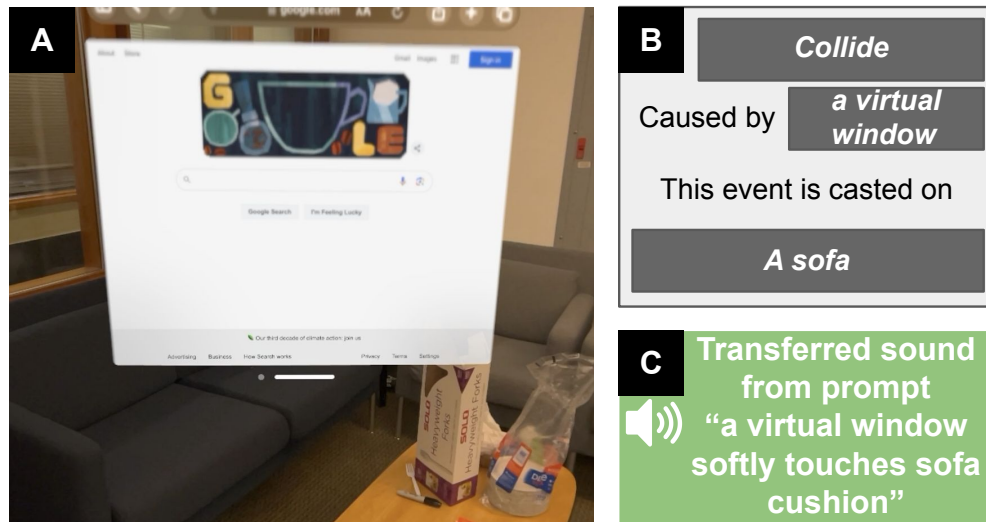


Figure 6.11: Sonification of window placement on Apple Vision Pro. (A) A user places a virtual window on a sofa. (B) Contextual information describes the action. (C) SonifyAR produces a text prompt and sound output.

We implemented a WoZ prototype based on Apple Vision Pro interaction (Figure 6.11). A Vision Pro user can adjust the position of application windows with hand gestures. In our scenario, SonifyAR generates a sound when a virtual window is placed on or intersects a physical surface such as a wall, desk, or floor. The pipeline processes these interactions with prompts such as “a virtual window bumps into a wall” and “a virtual window collides with a carpet.” The example illustrates how sound could communicate the relationship between virtual windows and surrounding surfaces.

6.2.5.5 Augmenting AR Authoring Processes

In addition to serving as a standalone PbD experience, SonifyAR could extend existing AR authoring tools. Adobe Aero and Apple Reality Composer already represent interactions through labels such as “Tap & Play Sound” and “Proximity & Jiggle.” These specifications could be sent to SonifyAR’s text-based pipeline,

allowing a plugin to provide matching sound assets for existing AR interactions.

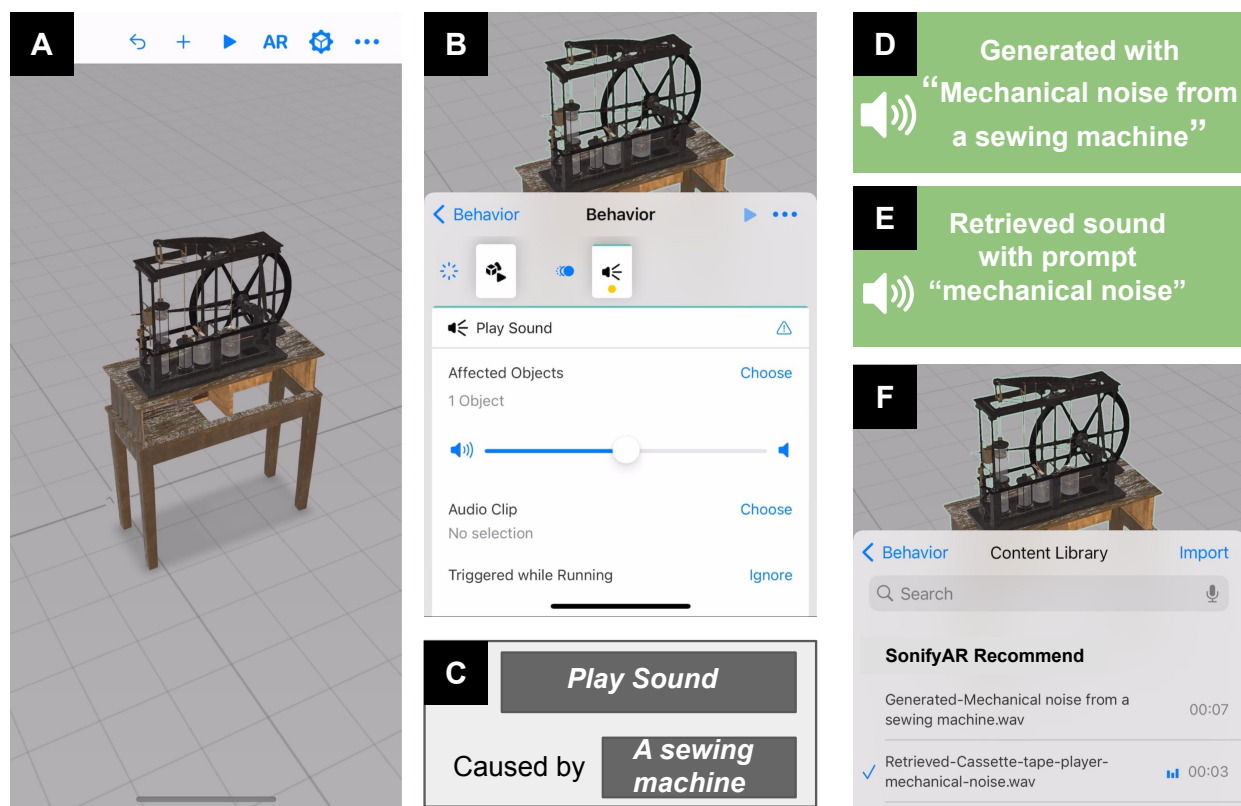


Figure 6.12: WoZ example of SonifyAR in the Reality Composer authoring process. (A) A machine model in the authoring interface. (B) Reality Composer’s interface for AR sound. (C) Contextual information for the event in B. (D, E) Sound assets generated and retrieved through SonifyAR. (F) The acquired sounds appear as recommended options.

We implemented an additional WoZ prototype using Apple’s Reality Composer and a 3D model of an animated machine (Figure 6.12). When authoring the model’s “Play Sound” behavior, the model description, trigger condition, and animation description can be sent to SonifyAR’s text-based pipeline. The resulting retrieved and generated sound assets appear at the top of the authoring interface under “SonifyAR Recommended.” The supplementary video shows the interaction in more detail.

6.2.6 Discussion and Future Work

In this project, we designed and implemented an AR sound-authoring pipeline that automatically acquires sound assets from contextual information. We examined opportunities in the AR sound-interaction space and implemented a custom pipeline that uses an LLM and generative AI to recommend, retrieve, generate,

and transfer sounds. We evaluated the pipeline through a user study and developed five proof-of-concept application scenarios.

6.2.6.1 Limitations and Failure Cases

Although participants responded positively to several aspects of SonifyAR, the study and application scenarios revealed limitations in its adaptability and output quality.

First, because the sound-acquisition pipeline relies on an LLM for both context interpretation and overall control, LLM errors can lead to failures. For example, the LLM may hallucinate context and send an inaccurate prompt to the sound-generation model, such as “sliding plastic on carpet” when a metal toy robot is sliding on a wooden floor. This produces a mismatched output.

Second, although AudioLDM [Liu et al., 2023] is a state-of-the-art text-to-audio generator, it can produce noisy outputs that do not match the prompt. This inconsistency is difficult to predict and cannot be fully resolved through prompt engineering.

Finally, some AR interactions, such as collisions and animated actions, are highly time-sensitive and require precise synchronization between sound and visual content. We currently use two synchronization methods. First, we provide example sounds for collisions, taps, and sliding interactions. Style transfer from well-timed examples helps the output remain synchronized with the AR interaction. Second, we bind colliders to 3D-model skeleton joints so that a sound plays only when the system detects a collision between an animated model and a physical surface. These methods do not apply to 3D assets without skeletons or to sounds unrelated to collisions, such as a dinosaur roar or mechanical-arm movement. Fully addressing synchronization between 3D animation and audio is beyond the current scope.

6.2.6.2 Future Work

We plan to address the following improvements in future work.

First, the current SonifyAR application supports only sound authoring and uses preauthored AR assets as input. To support a broader creation process, the sound-authoring pipeline should be integrated with tools such as Reality Composer and Adobe Aero. Future work also includes building a Unity SDK for professional AR developers and expanding the supported event types to categories such as “Spin,” “Emphasize,”

“Expand,” and “Jump,” which can be incorporated into SonifyAR’s text template.

Second, SonifyAR uses a straightforward text template to compile context from multiple sources, including virtual-object semantics and real-world semantics. It currently uses a single material-segmentation model to perceive real-world context. Future versions could incorporate more sophisticated computer-vision or multimodal foundation models to better understand both virtual objects and the real-world environment. Third, SonifyAR has basic error-handling features, including automatic removal of failed retrieval results and editing options for subpar sound assets (Section 6.2.3.3). Users can also correct recognition errors, such as incorrectly recognized materials, by specifying the correct material in text. Future versions could use a sound-quality inference model to validate outputs and automatically re-prompt the sound-generation model when needed.

Finally, SonifyAR’s sound-generation output presents opportunities for further refinement. The system is modular so that more advanced models can replace existing components as they become available. Each model, including the LLM, computer-vision models, and sound-generation model, can be updated independently.

6.2.7 Conclusion

In this project, we presented SonifyAR, a context-aware system for authoring sounds for AR events. Its custom PbD pipeline lets creators demonstrate an AR interaction, captures the interaction’s context, and uses AI-based recommendation, retrieval, generation, and transfer to acquire candidate sound assets. A study with eight designers provided initial evidence about the system’s usability and revealed limitations in generated-sound quality and interface feedback. Five application examples illustrated how the approach could extend to varied AR experiences. This research advances AR sound authoring and opens directions for applying LLMs and generative-audio models in interactive AR systems.

6.3 From Context-Aware Output to Manipulable Spatial Structure

SonifyAR shows how a structured event representation can translate sensed spatial context into another perceptual modality. The user evaluates the translation by listening in the same environment in which the event occurs. However, many spatial tasks require users not only to choose among generated outputs

but to inspect and reshape the spatial representation itself. Visual design from a single image is a useful example. People readily perceive surfaces, depth ordering, object extent, and body orientation in an image, yet conventional 2D tools do not expose that implicit geometry, while full 3D tools require substantial modeling expertise.

DepthScape extends the frontier from context-aware output to direct manipulation of AI-inferred structure. It reconstructs an image as a 2.5D scene, uses a vision-language model and modular geometry functions to identify meaningful objects and fit parametric planes, cylinders, spheres, faces, and skeleton-based anchors, and then exposes those anchors through mouse-following interactions. As in SonifyAR, the intermediate representation is relational and object centered. The difference is that it becomes an editable spatial handle rather than a textual description used to select a sound.

6.4 DepthScape: Parametric Anchors for 2.5D Visual Design

DepthScape investigates how AI-inferred depth can support accessible, expressive spatial authoring. A user begins with a single image rather than a manually constructed 3D scene. The system estimates a depth mesh, segments semantically meaningful regions, fits geometric primitives, and recommends parametric anchors. Users can inspect these suggestions, initialize a placement through direct manipulation, replace placeholder content, and refine position, scale, orientation, repetition, and occlusion before rendering the result.

6.4.1 Introduction

Humans gain 3D perception from 2D images through depth cues such as occlusion, perspective, and shading [Murray, 1994; Brenner and Smeets, 2018; Howard and Rogers, 2002]. Leveraging this visual ability, creators have long applied depth-enhancing techniques, such as layering and projecting, to boost visual impact and realism, leading to 2.5D visual designs. However, creating 2.5D visual designs can be challenging and arduous because designers must manually imitate depth cues, including relative ordering, occlusion boundaries, and perspective scaling, within flat 2D representations. This requires professional editing skills and iterative trial and error.

For example, to create a simple “surrounding figure” effect (Figure 6.14, panels B and C), the creator must simulate the perspective of a content ring and carefully layer it partly in front of and partly behind the

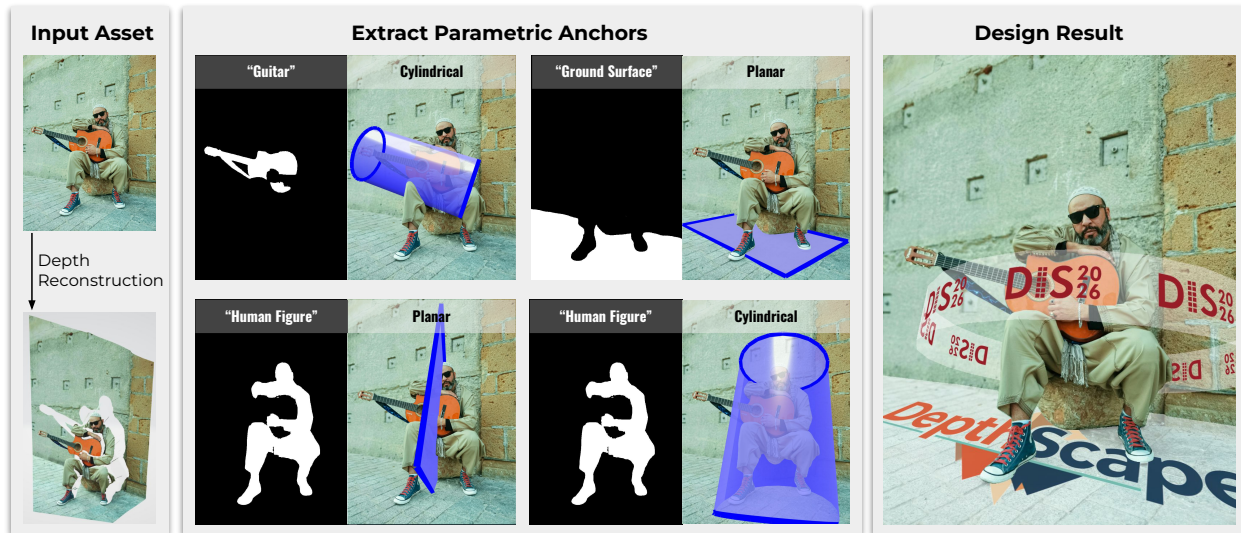


Figure 6.13: DepthScape reconstructs latent depth from a single image, extracts AI-recommended parametric anchors, and lets users place and manipulate design elements in the resulting 2.5D space.



Figure 6.14: Gallery of designs created with DepthScape. Icons in the top row show how each design is formed from one or more planar, cylindrical, or spherical parametric anchors.

figure. A creator can produce this effect in a 2D tool such as Photoshop or Illustrator, but must manually shape and align visual elements to match the implied 3D perspective. Even a small adjustment can require repeated refinement. Alternatively, the creator can construct a 3D scene in a tool such as Blender to obtain accurate depth cues and vanishing points. This workflow introduces a steep learning curve and the additional challenge of aligning a 3D camera and projection geometry with the source image. Both approaches shift attention from creative exploration to technical problem solving and make rapid experimentation difficult.

We introduce DepthScape, a human-AI collaborative authoring interface that bridges 2D editing with 3D rendering, enabling intuitive creation of 2.5D visual effects without requiring 3D expertise. DepthScape builds on two interaction innovations: (i) *depth-based blending*, which renders text and images in the source

image’s 3D depth scene to enable precise occlusion and perspective foreshortening; and (ii) *parametric anchors*, which bind content layers to objects in the original scene and thereby enable 3D placement through a 2D viewport. When the user uploads an input image, DepthScape uses monocular depth estimation [Wang et al., 2024a] to produce a 3D scene in which visual elements can be rendered with realistic occlusion and perspective aligned with the original image.

To further simplify 3D placement, DepthScape uses a vision-language model (VLM) [Hurst et al., 2024] to interpret semantic content and spatial structure and synthesize *visual programs* [Gupta and Kembhavi, 2023]. These programs guide 3D geometry extraction by masking key objects in the image [Ravi et al., 2024b] and fitting planes [Mariga], cylinders, spheres, or skeletons [Lugaresi et al., 2019] to them. The resulting parametric anchors align with objects in the input image and support direct-manipulation editing. Although this project focuses on a VLM-based pipeline, user input remains valuable and complementary. We therefore position the approach as a first step toward a hybrid human-AI pipeline for anchor generation.

We created DepthScape through a human-centered, iterative design process. We initially developed a proof-of-concept prototype to test the feasibility of depth-based blending and AI assistance for 2.5D design. We then conducted a formative study with nine participants to observe usage patterns, assess usability, explore potential applications, and gather feedback on interaction methods. Participants replicated example designs and created open-ended designs. Their experiences showed how recommendations could help novices explore possibilities and how rapid parameter changes could help experienced designers test ideas. The study also identified needed improvements in rendering, AI recommendations, and editing interactions. These findings informed the final DepthScape system, which automatically extracts parametric anchors and enables 3D placement through direct manipulation in a 2D viewport.

To evaluate the final DepthScape prototype, we performed a technical evaluation with 100 professional stock photographs spanning several types of visual content. We processed each image with DepthScape and logged processing time, errors, visual programs, and extracted anchors to characterize the pipeline’s efficiency, robustness, and output diversity. We also conducted an expert review with three professional designers, who rated the suggested anchors and design outputs and identified opportunities for improvement. Finally, we demonstrate the range of the approach through five application scenarios, including 2.5D video creation, modification of photographed scenes, and storyboarding with hand-drawn scenes.

Our contribution is threefold. First, we propose a depth-based blending pipeline for 2.5D visual design that places visual elements in monocular depth reconstructions to produce occlusion and perspective foreshortening. Second, we use a VLM to generate visual programs that extract semantically grounded parametric anchors and enable direct-manipulation editing. Third, we characterize the approach through a formative user study, a technical evaluation, an expert review, and five application examples.

6.4.2 Designing DepthScape

Building on the core idea of depth-based blending, which places content in reconstructed depth spaces, we conducted a three-stage iterative process to test feasibility, explore user needs, and discover key interaction patterns. We first built a proof-of-concept prototype. We then used it in a formative user study with nine participants of varying design expertise. Finally, based on the study findings, we developed the final DepthScape system with improved depth reconstruction, interaction, and AI design suggestions.

All DepthScape studies involving human participants, including the formative study and expert evaluation, were reviewed and approved by the University of Washington Institutional Review Board (UW IRB; STUDY00022751).

6.4.2.1 Proof-of-Concept Prototype

We built a proof-of-concept web prototype (Figure 6.15). The prototype uses the single-image-to-3D-mesh model CRM [Wang et al., 2024c] to reconstruct the main object of a user-uploaded image as a 3D mesh. We then add primitives such as planes, cylinders, and spheres to render realistic occlusion and perspective effects. Users can edit the placement of these primitives with sliders that adjust position, rotation, scale, and content placement on primitive surfaces. We also explored AI recommendations for 3D placement by leveraging CLIP [Radford et al., 2021b] and cosine similarity to retrieve placement parameters from similar input images. The recommended parameters are rendered as thumbnail images for users to browse and select.



Figure 6.15: Proof-of-concept prototype interface.

6.4.2.2 Formative User Study with the Prototype

To test the feasibility, usability, and helpfulness of our proposed 2.5D creation pipeline, we conducted a formative study with nine participants between 20 and 60 years old. To examine the experiences of people with different levels of expertise, we recruited four design enthusiasts and five professional designers with varying backgrounds in 2D and 3D design. We introduced the prototype and guided participants through recreating two example designs. Participants then explored the system freely and created designs of their choosing. We concluded with usability questions and rating scales. Our prior work reports the complete study protocol and findings [Su et al., 2025b].

All participants used the prototype to create multiple 2.5D designs. They gave the statement “the prototype supported my creativity” a mean rating of 4.67 out of 5. They also rated the ease of creating occlusion and surrounding effects at a mean of 4.56 out of 5 for each. The open-ended task produced diverse designs (Figure 6.17) .

The formative study identified application scenarios centered on rapid prototyping and ideation for 2.5D effects. P8 described using the tool for “trying out ideas at early stage,” obtaining “a few AI brainstorm suggestions,” and creating “rough mock ups” before moving to higher-fidelity tools. P1 said that it made “occlusion and surround effects easier and faster” than workflows that required knowledge of 3D software. These responses position DepthScape as a specialized environment for exploring pseudo-3D design possi-



Figure 6.16: In the formative user study, participants replicated two designs (left) using the provided assets (middle) and the proof-of-concept prototype. Example results are shown on the right.



Figure 6.17: Design results of the open-ended exploration of the formative user study.

bilities before moving to production tools such as Photoshop or 3D software.

Amateur and professional designers valued the tool in different ways. Amateur designers emphasized the freedom to experiment; P9 said that it “enables you to do a lot of different stuff.” The depth cues and exploratory controls allowed them to create effects without specialized 3D knowledge. Professional designers evaluated the tool in relation to established workflows and quality standards. They appreciated its distinctive controls but wanted tighter integration with existing tools. As P6 remarked, “(It’s okay that) We don’t have all those text editing capabilities because every other tool already has that. We don’t want to rebuild the wheel.” Thus, amateurs treated DepthScape as an approachable entry point to 2.5D design, while professionals saw it as a potential complement to advanced toolchains.

From these design explorations, we identified three ways in which the prototype supported participants’ work: (1) depth reconstruction simplified the creation of occlusion effects; (2) rendering deformation and repetition in 3D supported complex but orderly perspective effects; and (3) the placement parameters created a sufficiently broad space for participants to encounter unplanned design variations.

6.4.2.3 Improvements Based on Study

Alongside the positive findings, we identified several aspects that required improvement. The following findings guided the next system iteration.

From Orthographic to Perspective Rendering. Because of CRM’s reconstruction behavior [Wang et al., 2024c], as well as that of many existing depth-estimation models [Yang et al., 2024a,b; Patni et al., 2024; Zhu et al., 2024; Xu et al., 2024b,a; Zhao et al., 2024], its reconstructions matched the input image only under orthographic rendering. Our initial prototype therefore used an orthographic camera for the design canvas. This choice eliminated perspective foreshortening, and many participants found the depth effects difficult to perceive and less visually impactful. We switched to perspective rendering for both the depth reconstruction and the added visual elements.

From Main-Object to Whole-Scene Reconstruction. The reconstruction model in the prototype captured only the primary object in an input image, limiting its use to single-object compositions. It could not support more complex urban, natural, or tabletop scenes. To accommodate a broader range of images, the

final system uses a model that reconstructs depth across the complete scene.

From Sliders to Direct Manipulation. Many participants, especially those with less 3D design experience, found it difficult to place elements with parameter sliders. First, unfamiliar parameters made editing outcomes difficult to anticipate, leading to repeated trial and error. Second, a conceptual goal such as surrounding a bullet with a ring of text (Figure 6.16B) required the coordinated adjustment of several 3D parameters. Participants preferred direct-manipulation controls that corresponded to design intentions and image semantics. We therefore developed controls based on parsed image content and depth geometry. For an image containing a bullet, for example, the system can suggest a cylindrical placement aligned with the bullet and expose the cylinder radius as a directly manipulable parameter.

From Parameter Recommendations to Parametric Anchors. Our initial exploration of AI recommendations for 3D placement parameters was not sufficiently helpful (mean = 3.67/5) because retrieval-based parameter matching was imprecise with respect to image semantics and depth geometry. Visual designs are highly sensitive to layout, so small mismatches between image content and 3D placement can disrupt the intended effect. We therefore use a VLM to parse the input image semantically and extract geometry directly from the depth reconstruction, improving placement precision through parametric anchors that bind content layers to image objects.

6.4.3 The DepthScape System

Building on the iterations above, we introduce DepthScape, a human-AI collaborative authoring pipeline that supports 2.5D graphic design through depth estimation, semantic interpretation, and geometry extraction. DepthScape uses the monocular geometry-estimation model MoGE [Wang et al., 2024a] to reconstruct a 3D mesh that follows the visual depth cues in an input image. This reconstruction creates a 3D space for 2D design elements such as text and images, enabling depth-based blending with occlusion and perspective effects. A VLM further assists the process by identifying visual components and suggesting parametric anchors that support direct-manipulation placement.

6.4.3.1 Depth-Based Blending

We use MoGE [Wang et al., 2024a] to create a 3D mesh that reflects the visual depth cues in a 2D image. Unlike many existing depth-estimation models, MoGE performs affine-invariant monocular geometry estimation and produces reconstructions that align with input images under perspective rendering. The reconstructed mesh encodes depth cues and provides the foundation for our authoring pipeline.

We then implement depth-based rendering by placing both the reconstructed depth mesh and any additional 2D visual elements in a shared 3D space (Figure 6.18, left). By default, rendering this 3D scene correctly produces occlusions between the depth mesh and the added 2D content. However, it also exposes the reconstructed depth mesh itself, whose visual quality is noticeably lower than that of the original image. To address this, we make the depth mesh invisible while preserving its occlusion effect. Specifically, we disable color-pixel writing for the depth mesh, turning it into an invisible object that blocks other elements and leaves holes in the rendered image. We fill these holes using pixels from the original input image so that the final output appears as if objects in the original image occlude the added visual elements (Figure 6.18, right).

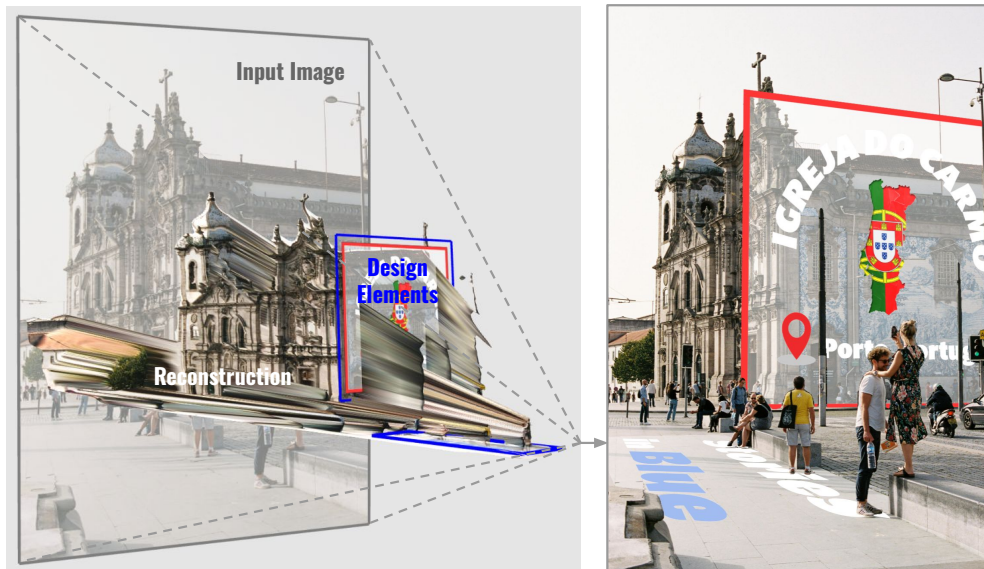


Figure 6.18: DepthScape’s 3D-space composition and rendering results. Left: a 3D space viewed from a different angle, showing the original input image, reconstructed point cloud, and added design elements. The gray dashed line indicates the camera position. Right: rendered results with design elements embedded in the input-image scene.

6.4.3.2 Parametric Anchors

During the formative study, we observed challenges in precisely placing design elements in reconstructed 3D spaces. We also identified two patterns that could simplify the task. First, design elements were often anchored to objects in the input image, such as text wrapped around an arm or a plane aligned with a building facade. Second, designers explored variations by adjusting one or two key parameters while keeping the others fixed to preserve the design relationship, such as shifting a plane along its normal vector while keeping it parallel to a building.

To formalize these patterns, we borrow the concept of *anchors* from augmented reality and define *parametric anchors* that bind visual content to objects in an image. Each anchor aligns content with selected geometric properties of an object, such as a plane parallel to a building facade, a cylinder surrounding a human figure, or a sphere centered on an owl (Figure 6.19). Parametric anchors constrain placement to the most relevant degrees of freedom. A planar anchor has a fixed reference plane and moves only along its normal. A cylindrical anchor organizes placements around a shared central axis and exposes radius and height adjustments. A spherical anchor centers placements on a fixed point and varies in radius.

By leveraging these parametric anchors, we simplify unconstrained 3D placements into structured adjustments that align with scene geometry. This enables direct manipulation in a 2D viewport, which would otherwise be ambiguous. For instance, translating a plane freely in 3D from a 2D perspective is ill-defined, but within a Planar parametric anchor, it reduces to a simple 1D adjustment along the normal vector, which can be directly mapped to mouse input. See Figure 6.19 for examples.

6.4.3.3 Extracting Parametric Anchors Through Code Synthesis

Extracting parametric anchors remains challenging. Heuristic methods struggle in open-vocabulary settings to integrate object type, shape, and spatial layout. Manual approaches require users to understand both anchors and the intermediate extraction steps. To explore automatic extraction, we use a VLM, GPT-4o [Hurst et al., 2024], to parse the input image and guide geometry extraction. Inspired by prior work on visual programming [Gupta and Kembhavi, 2023], we prompt the VLM to follow a visual-program template that supports modular, sequential extraction focused on selected image components. Each generated program encodes a design intention as a sequence of the following geometry functions:

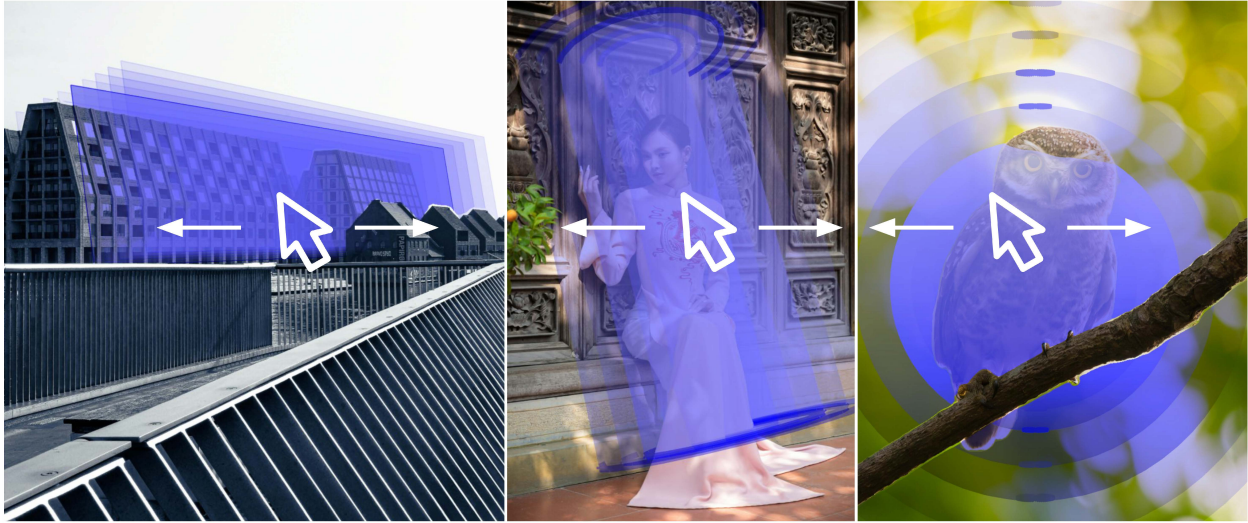


Figure 6.19: DepthScape currently supports three types of parametric anchors that can be directly manipulated through mouse movement. From left to right: planar, cylindrical, and spherical.

Segment the Input Image. The `Text2Mask(prompt)` function segments the input image using a text prompt generated as part of the visual program. The backend uses Grounded SAM 2 [Ravi et al., 2024b] to produce a corresponding mask, for example from the prompt *the train in the image*. The `Mask2Pointcloud` function then selects vertices of the reconstructed depth mesh that fall inside the mask. The resulting point cloud becomes input to subsequent geometry extraction.

Extract 3D Geometry. We implement several geometry-extraction functions, which the VLM selects based on image semantics.

`Pointcloud2Plane` fits planes to point clouds for planar objects such as building facades, ground and water surfaces, and tabletops. It uses PyRANSAC-3D [Mariga]. The function also derives properties such as the primary direction and perpendicular extrusion, which support geometric relationships among objects.

`Pointcloud2Cylinder` fits a containing cylinder to an elongated object such as a human figure, tree trunk, or train. The function accepts an optional axis direction. When none is provided, it uses principal component analysis to estimate the axis from the point cloud, then finds the center and smallest radius that contain the selected points.

`Pointcloud2Sphere` fits a sphere to a rounded object by estimating a center and the minimum radius needed to contain the point cloud.

In addition to these primitive-fitting functions, we implement functions that extract human pose and facial structure.

`Pointcloud2Skeleton` uses MediaPipe [Lugaresi et al., 2019] to extract 2D pose landmarks and projects them onto the depth point cloud to obtain 3D skeleton points. The function derives frontal and median planes for content placement, as well as upward and front-facing directions that other functions can use.

`Pointcloud2Face` extracts geometry for close-up faces. As in skeleton extraction, the function uses MediaPipe [Lugaresi et al., 2019] to detect 2D facial landmarks and projects them onto the depth point cloud. The resulting 3D face points define frontal and median planes and upward and front-facing directions.

Create Parametric Anchors. The final functions convert extracted planes, cylinders, spheres, and directions into parametric anchors.

`Planar(plane)` creates a planar parametric anchor from an extracted or derived plane and optional directions.

`Cylindrical(cylinder)` creates a cylindrical parametric anchor from an extracted cylinder.

`Spherical(sphere)` creates a spherical parametric anchor from an extracted sphere and optional direction.

By combining these functions, DepthScape extracts geometry from a depth reconstruction according to image semantics. Figure 6.20 shows a visual program, its intermediate steps, and the resulting parametric anchor.

We prompt the VLM with 16 example visual programs to establish the program syntax and general design patterns. Each example includes a raw input-image description and a matching visual program. The VLM then generates semantically reasonable visual programs for new input images. DepthScape parses the generated programs and runs the extraction pipeline to create parametric anchors. Successfully extracted anchors, together with the original design rationale, are shown in the interface for user interaction.

6.4.3.4 The DepthScape Interface

The DepthScape interface is implemented as a web application with React and Babylon.js (Figure 6.21). It has three main parts: (1) the asset panel imports input images, shows suggested parametric anchors, and lists

```

"name": "Train Structure" ,
"description": "Extracts the main train structure from the image to
create a cylindrical anchor." ,
"visual_program": [
    "MASK_0=Text2Mask(prompt = \"the train in the image\")" ,
    "POINTCLOUD_0=Mask2Pointcloud(mask = MASK_0)" ,
    "CYLINDER_0=Pointcloud2Cylinder(pc = POINTCLOUD_0, direction =
NULL)" ,
    "CYLINDRICAL_0=Cylindrical(cylinder = CYLINDER_0)" ]

```

A

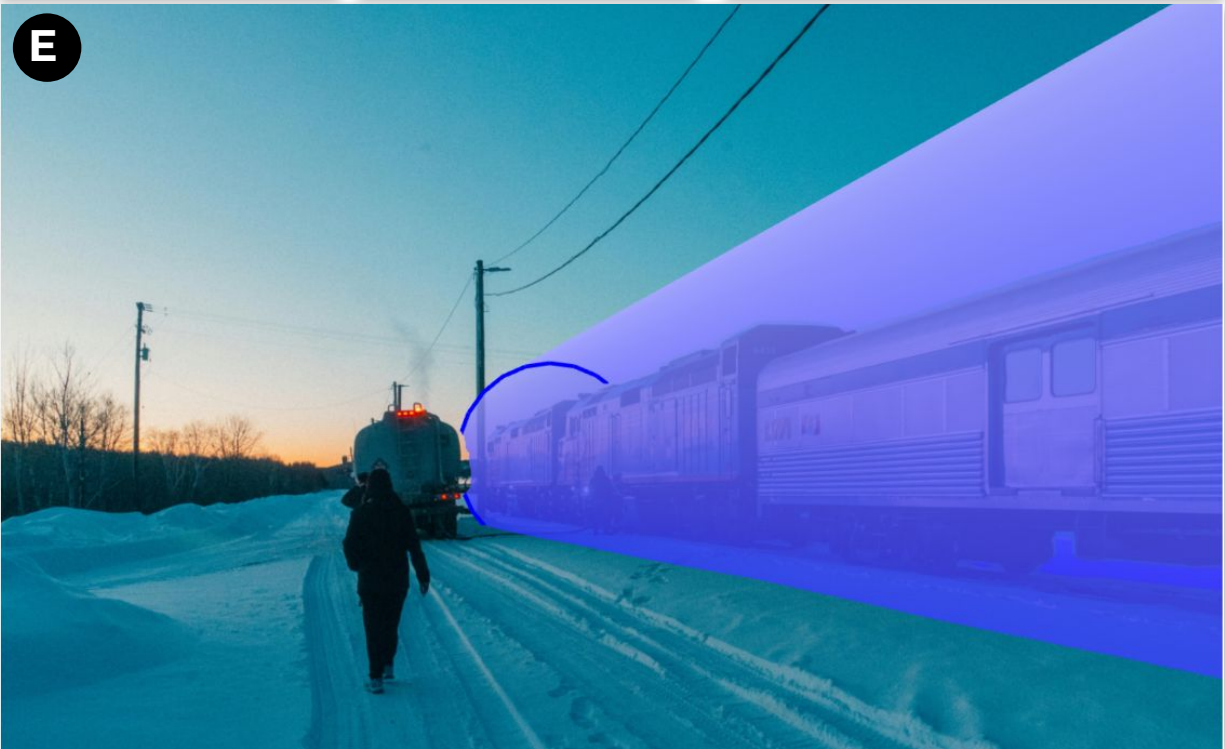
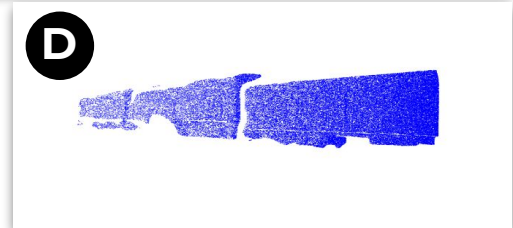
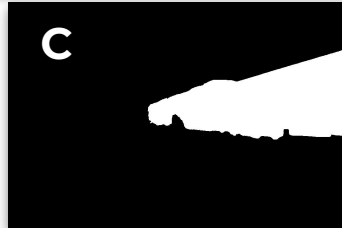


Figure 6.20: DepthScape employs a vision-language model to (A) generate visual programs from (B) input images. The program guides a cylindrical-anchor extraction process, including (C) masking the prompt the train in the image and (D) selecting the corresponding point cloud. The final result is (E) an interactive cylinder precisely aligned with the train.

added scene layers; (2) the canvas renders 2.5D effects and enables direct manipulation of design elements; and (3) the editing panel lists options for selected parametric anchors, including adding text or image content and adjusting placement parameters.



Figure 6.21: DepthScape’s user interface. (A) The upload control and input image. (B) AI-suggested parametric anchors. (C) Layers added to the scene, with controls to select, hide, show, or remove them. (D) The direct-manipulation canvas. (E) The parameter panel during layer initialization. (F) Additional controls available after initialization, including content rotation and the addition of (G) image content. (H) Fine-tuning controls for layer parameters.

The user first imports an image (Figure 6.21A). DepthScape reconstructs its depth mesh and extracts suggested parametric anchors. The user can preview an anchor by hovering over its list entry (Figure 6.21B) and can select a suitable option. Selection begins an initialization sequence that offsets the anchor and, for planar anchors, resizes its surface, centers the content’s UV coordinates, and scales the content UVs. Each step uses mouse-following direct manipulation. After initialization, buttons and a slider allow the user to rotate and mirror the 2D content (Figure 6.21F, lower half). An advanced panel exposes anchor position, scale, and orientation so that users can correct inaccuracies and customize the result (Figure 6.21H). A white “LOREM IPSUM” placeholder helps users assess the initial placement; they can replace it with typed text or an uploaded image (Figure 6.21G). A single command exports the final result as a rendered image. The supplementary video provides a complete interaction walkthrough.

6.4.4 Evaluation

We evaluated DepthScape in terms of processing efficiency, robustness, output diversity, and design quality. We processed a varied image collection and examined processing time, errors, the distribution of AI-suggested parametric anchors, and expert ratings of anchor and output quality.

6.4.4.1 Collecting Test Images

To obtain a varied selection of test images, we collected images from Pexels [Pexels, 2025], a free stock-image platform. We sampled its trending-image feed on two dates, collecting 50 images on each date for a total of 100. The collection contains four content types: portraits, urban or building scenes, nature or animal scenes, and still objects. Figure 6.22 shows the distribution and examples.



Figure 6.22: Four types of test images.

6.4.4.2 Procedure

We sequentially uploaded all 100 images to DepthScape and inspected the processing time, VLM-generated visual programs, extraction result for each program, and error logs. We analyzed these logs to characterize system efficiency, robustness, the distribution of parametric-anchor types, and causes of failure.



Figure 6.23: Example design results shown to experts.

We also evaluated suggested-anchor and final-output quality through an expert review with three professional designers. Table 6.1 reports their backgrounds. We showed a recording of the DepthScape interface and recommended anchors for four images, one from each content type. We also presented completed designs to illustrate output quality (Figure 6.23). The experts rated the quality of the anchors and design outputs on seven-point scales and discussed potential applications and system improvements.

Table 6.1: Expert-review participant demographics.

Expert	Gender	Age	Design experience (years)	Design specialties
E1	Male	25-34	1.5	UI/UX
E2	Female	25-34	12	Game, UI/UX
E3	Male	45-54	25	Web, Animation, Motion Graphics

6.4.4.3 Technical Results

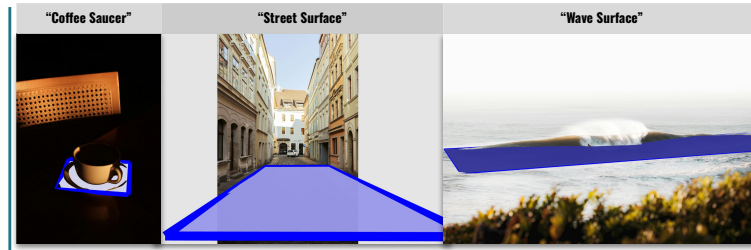
DepthScape successfully processed 97 of the 100 test images and generated at least one semantically grounded parametric anchor for each. An image took 21.1 seconds on average to process ($SD=5.5$ seconds) and yielded a mean of 4.2 visual programs ($SD=1.1$). Eighty-nine percent of generated visual programs produced parametric anchors available for user interaction.

Figure 6.25 shows the mean distribution of processing time. Waiting for GPT responses accounted for 63.7% of total processing time ($M=14.06$ seconds, $SD=4.04$ seconds), making model latency the largest source of delay. Depth reconstruction accounted for 8.4% ($M=1.86$ seconds, $SD=2.02$ seconds), masking for 13.1% ($M=2.89$ seconds, $SD=1.23$ seconds; per-mask $M=0.89$ seconds, $SD=0.19$ seconds), and geometry extraction for 14.8% ($M=3.28$ seconds, $SD=1.30$ seconds).

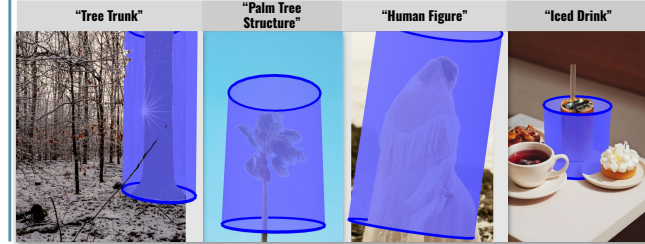
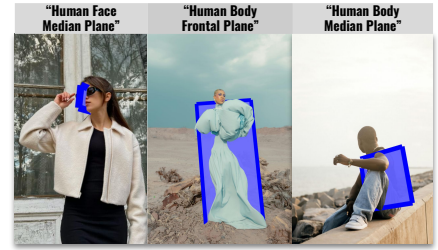
The VLM generated 419 visual programs, of which 370 completed geometry extraction and yielded interactive parametric anchors. Figure 6.24 shows the distribution of anchor types. Planar anchors were the most common, including 180 object planes (42.96%) and 113 skeletal or facial planes (26.97%). The VLM also suggested 94 cylindrical and 28 spherical anchors.

We analyzed 46 failed planar visual programs and identified two error types. In 34 cases, incomplete or failed masks of human figures prevented detection of skeletal or facial landmarks. In the other 12 cases, the VLM produced an invalid program, such as one containing a nonexistent function or lacking a valid

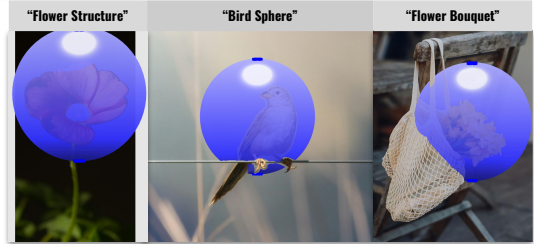
Object Planes 180 (42.96%)



Human Planes 113 (26.97%)



Cylindrical 94 (22.43%)



Spherical 28 (6.68%)

Figure 6.24: Four types of parametric anchors. Each example shows the name of the parametric anchor on top.

Process Time Distribution

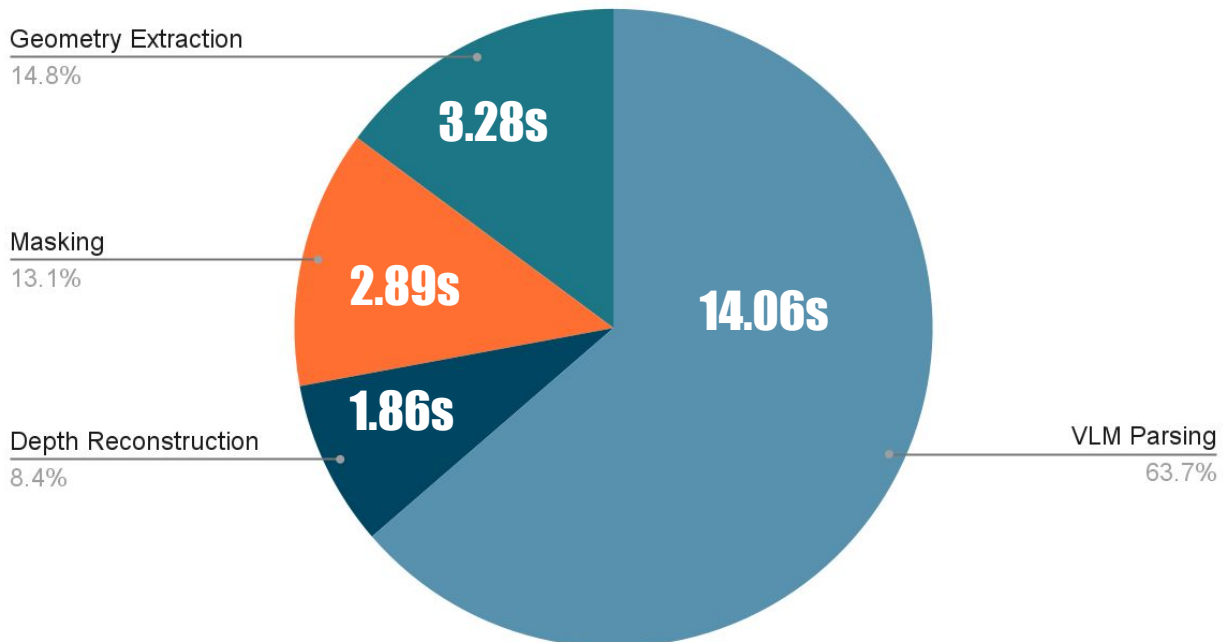


Figure 6.25: The distribution of processing time.

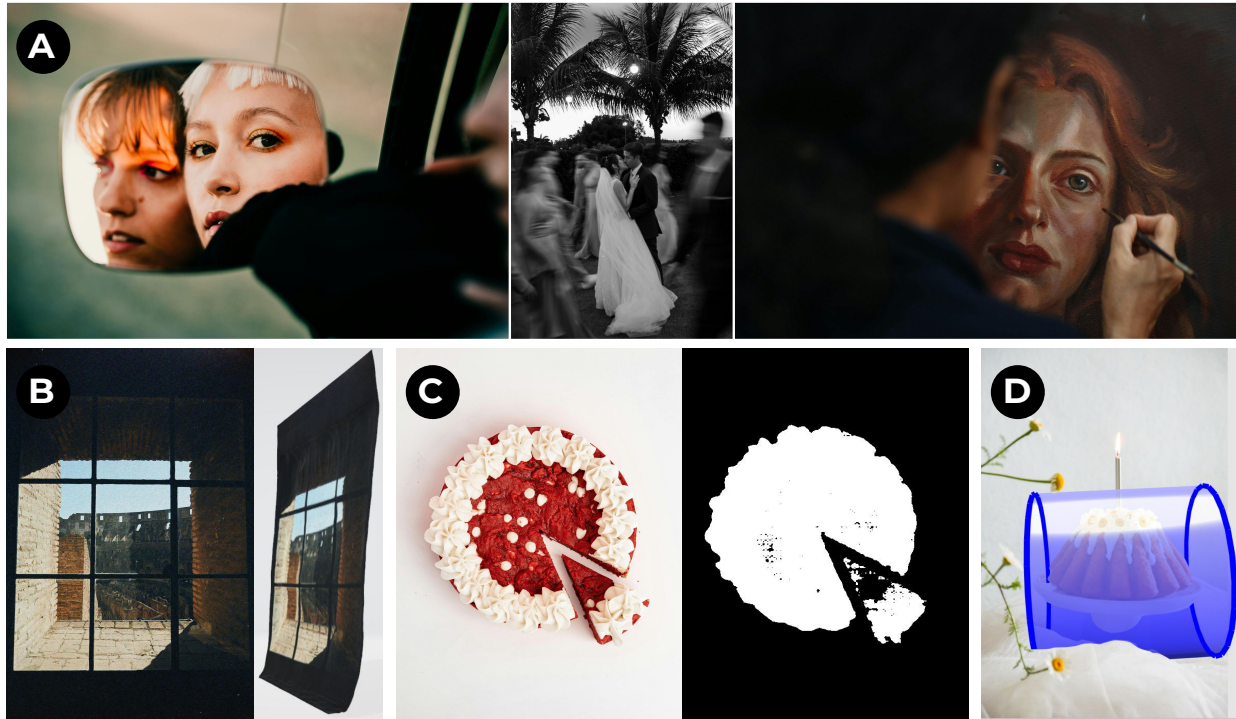


Figure 6.26: Failure cases. (A) Three images for which DepthScape generated no parametric anchor. (B) Reconstruction of a transparent glass surface loses the geometry behind the glass. (C) A request to mask “the cake slice” selects the entire cake. (D) Cylinder fitting selects a horizontal axis instead of the shorter, semantically appropriate vertical axis.

parametric-anchor output. Figure 6.26A shows three images in which human-figure extraction failed.

We also observed failures that produced unsuitable parametric anchors. In Figure 6.26B, reconstruction flattened a glass surface and lost the geometry visible behind it. The VLM nevertheless suggested fitting a plane to the ground behind the glass, producing a misaligned result. In Figure 6.26C, the VLM requested a mask for “the cake slice,” but the segmentation model selected the entire cake. In Figure 6.26D, the cylinder-fitting module selected a horizontal axis for the cake rather than the shorter but more semantically appropriate vertical axis. Beyond these geometric errors, occluded or warped text layers can be difficult to read. As Figure 6.23 illustrates, the balance between 2.5D realism and readability depends on the text and varies across designs. Designers must therefore consider text layout, content, and the degree of occlusion.

6.4.4.4 Expert Review Results

The three experts rated parametric-anchor quality as 7, 4, and 5 on a seven-point scale and design-output quality as 6, 4, and 5. They appreciated DepthScape’s ability to parse complex scenes and generate varied anchors based on image content. E1 remarked, “I can totally imagine that a Vogue cover designer would use it very powerfully.” E2 emphasized its value to novices: “I think the value of this is that 99% of people who have no idea how to do this [2.5D effects] will benefit a lot from this.” The experts also identified improvements needed for professional use, including occlusion precision, rendering quality, and perspective accuracy. E3 observed that although cylinders and spheres were effective, plane edges were sometimes slightly misaligned with the underlying objects and required manual correction.

The experts proposed several application scenarios. E1 suggested book covers, T-shirts, and memes and envisioned generating 2.5D animations by binding a position parameter to time. E2 emphasized its value for design ideation: “I think it’s nice to use [DepthScape] to quickly put together some concepts. It’s definitely helpful to accelerate the workflow.” They also noted its potential for people using platforms such as Canva or Instagram who want to create striking text effects without expertise in software such as Photoshop. E3 suggested interface mockups, presentation title slides, promotional materials, and video effects that use both semantic and depth analysis.

Experts also wished for additional features. E3 hoped for more sophisticated content anchors that incorporate multiple directional cues from the scene. They also emphasized the need for text readability enhancements when adding text elements. E2 envisioned DepthScape being integrated into existing design platforms like Figma, Canva, or Instagram, to streamline content creation. All three experts requested improved blending between the depth scene and added content, including support for light estimation, shadow casting, and blur effects to further enhance realism. Additionally, E2 and E3 advocated for the ability to export designs in editable, multi-layer formats (e.g., PSD files) for further refinement in professional design software.

Overall, experts viewed DepthScape as a promising and approachable tool for expressive 2.5D design, particularly during rapid ideation and creative exploration. Their feedback also highlights a gap between current capabilities and the precision required in professional workflows. Future improvements should therefore focus on geometric accuracy, visual integration, and editing control.

6.4.5 Application Scenarios

Inspired by the formative study and expert review, we implemented five application scenarios to illustrate DepthScape’s range and potential. The supplementary video provides additional details and animations.

6.4.5.1 Integration into Image Editing Tools

Participants in both the formative study and expert review suggested integrating DepthScape with existing image-editing tools. A plugin could make depth-based blending available within those platforms, while export to layered, editable files could allow a designer to begin in DepthScape and continue refining the result in another tool. Either approach would preserve DepthScape’s specialized 2.5D controls while connecting them to broader production workflows.

6.4.5.2 2.5D Video Effects

DepthScape’s effects can extend from static images to video in two ways suggested by study participants. First, a placement parameter can vary over time to animate content in a static image, such as rotating an arrow ring or moving a text plane. Second, DepthScape could parse consecutive video frames and apply consistent parameters for object-centered editing. The supplementary video illustrates both approaches.

6.4.5.3 Editing Photographs of Physical Scenes

Because DepthScape approximates depth cues and blends additional visual elements into an image, it can support rapid previews of modifications to a physical scene. For example, a designer can add images of rugs or paintings to preview an interior design or place poster graphics in a street photograph to test branding concepts (Figure 6.27).

6.4.5.4 Simulating AR Scenes

By anchoring content to physical objects and surfaces, DepthScape can blend virtual assets, such as interface panels and 3D models, into a photograph without tracking the environment. This capability supports prototyping an AR composition from a single image. For example, planes aligned with the floor and walls



Figure 6.27: DepthScape adds content, including a rug, wall art, and advertising graphics, to photographs of physical scenes.

can support virtual panels (Figure 6.28), while other anchors can place animated 2D or 3D content in the scene.

6.4.5.5 Storyboarding

Hand-drawn scenes can encode relative depth cues that DepthScape reconstructs as approximate 3D geometry. The system can therefore blend 2D assets into a sketch for rapid storyboarding, allowing creators to combine rough line work with more polished graphics (Figure 6.29).

6.4.6 Discussion and Future Work

In this project, we presented DepthScape, a human-AI collaborative authoring tool that lets 2D designers create 2.5D visual effects by arranging assets within a *monocular depth reconstruction*. Designers work through a 2D canvas while the system uses a 3D scene to render depth effects. DepthScape’s AI pipeline connects RGB imagery with reconstructed geometry: it analyzes an input image, synthesizes geometry-extraction code, and executes that code to generate semantically grounded parametric anchors for editing and exploration. We evaluated DepthScape through a formative study with nine participants, a technical evaluation with 100 stock images, an expert review with three professional designers, and five exploratory

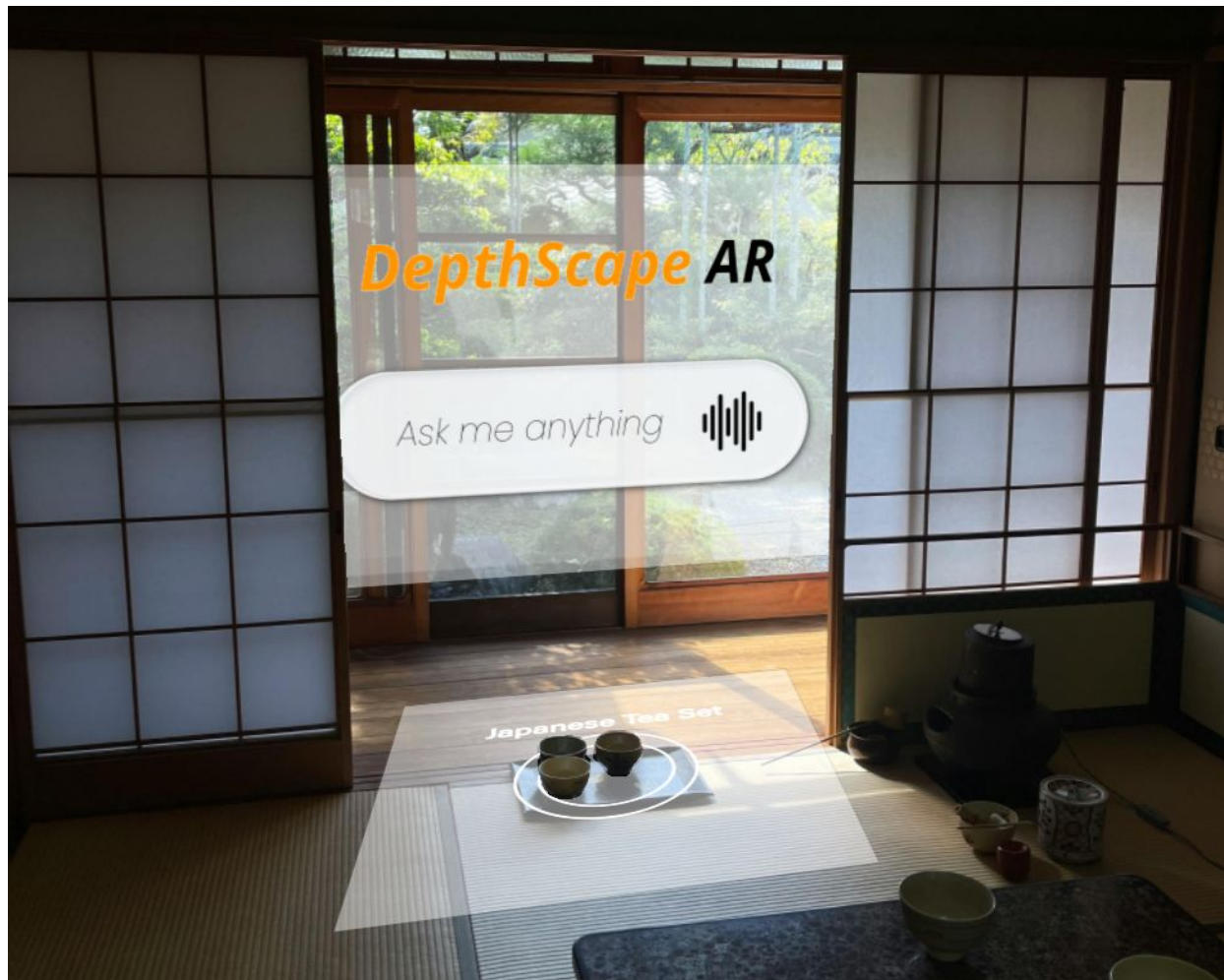


Figure 6.28: Simulated AR effects, including the plane and text for the tea set, and also the virtual UI located behind the sliding doors, in a real-world indoor scene.

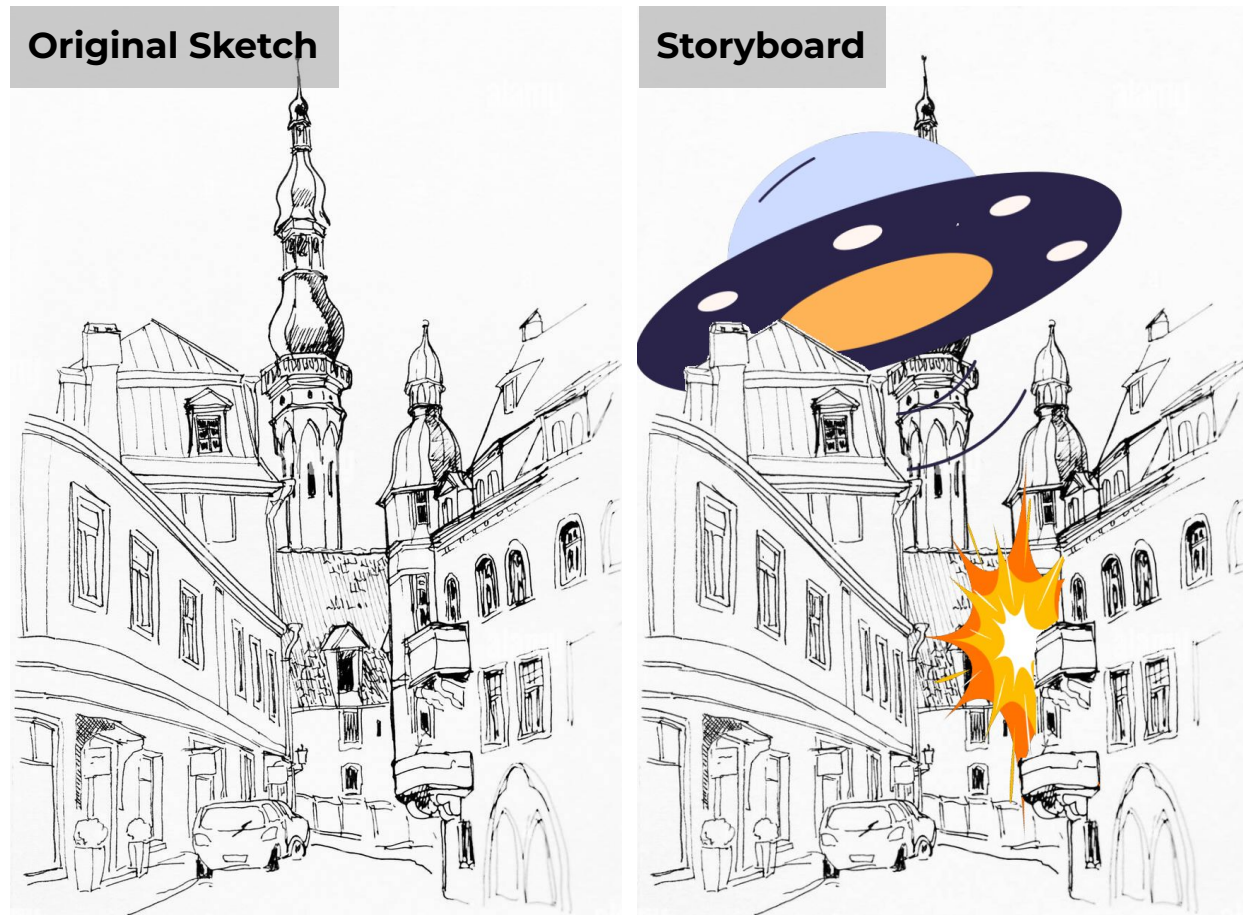


Figure 6.29: DepthScape applies to sketched scenes, supporting rapid storyboarding with visual assets of varying fidelity.

application scenarios.

DepthScape occupies a middle ground among 2D, 3D, and generative editing workflows. Participants used it to blend 2D assets in a pseudo-3D space while retaining direct control over placement parameters. Traditional 2D editors such as Photoshop can produce similar effects but require manual adjustments to perspective and occlusion. Three-dimensional tools such as Blender render these cues directly but require 3D scene construction and camera alignment. Generative image editors, such as Gemini 2.5 Flash Image¹, can synthesize broad image changes but offer a different form of control and may modify source pixels that a designer wishes to preserve. DepthScape complements these approaches by combining direct manipulation of added content with depth-aware rendering over the source image. The current studies establish its value

¹<https://aistudio.google.com/models/gemini-2-5-flash-image>

for exploration and rapid ideation, but do not directly compare it with these established tools.

6.4.6.1 Semantics and Geometry Extraction

DepthScape’s geometry extraction currently handles primarily single objects and simple geometric types. Future work could enable more complex compositions. For example, combining the front-facing direction of a human figure with the normal of a ground plane could produce a richer planar anchor. The visual-program format can express such multi-element relationships, but reliable synthesis will require additional examples and evaluation.

Our current system supports planes, cylinders, and spheres as anchor primitives, informed by common 2.5D design patterns observed in our design research and findings from our formative study. Nevertheless, certain creation needs fall outside this taxonomy, such as decal effects that place content on irregular surfaces (e.g., cloth), more complex geometries (e.g., cones or tori), or free-floating content not associated with any object. While our interface currently allows users to manually adjust placement of primitives via parameter controls, extending support to richer geometry types is an important direction for future work.

6.4.6.2 Advanced Blending Effects

DepthScape uses depth-based blending to enable realistic occlusion and perspective cues, but expert feedback points to opportunities in richer physical effects. Shadows, reflections on water or glass, color bleeding, fog, depth-of-field blur, and motion blur could further blend the inserted content into the scene. We plan to enable these effects in future iterations by estimating additional environmental cues and upgrading the rendering pipeline.

6.4.6.3 Limitations

Despite these contributions and future directions, several limitations remain in this project.

Evaluation design. We did not conduct a direct comparison study against industry tools such as Photoshop or Blender. DepthScape is currently tailored to 2.5D effects rather than the broader image/3D editing capabilities of these established platforms. At the same time, existing 2D, 3D, and generative approaches support different workflows and affordances, making it difficult to design fair and meaningful cross-platform

tasks for comparative evaluation. Future work could develop task-specific benchmarks that highlight complementary strengths and situate DepthScape more clearly within the ecosystem of design tools.

System performance. Processing one image currently takes approximately 21 seconds, most of which is spent waiting for VLM responses. Smaller task-specific models, cached or incremental inference, and parallel execution of independent visual programs could reduce this latency.

User interaction. Although the current design provides VLM-selected parametric anchors and direct-manipulation editing, expert users requested finer controls, such as explicitly selecting target objects, specifying anchor types, and manipulating more sophisticated geometries such as splines. Future versions should also expose intermediate AI outputs underlying automatically generated anchors, including image masks, geometry-extraction steps, and the VLM’s proposed program. These representations could help users understand how an anchor was derived, identify errors, and revise the result according to their preferences.

6.4.7 Conclusion

We presented DepthScape, a graphic-authoring tool that combines depth estimation and vision-language models for 2.5D design. DepthScape reconstructs a 3D scene from a 2D input image and places visual elements in that scene to produce depth-based occlusion and perspective effects. A VLM synthesizes visual programs that extract parametric anchors from image semantics and reconstructed geometry, allowing users to place content through direct manipulation and fine-grained parameter editing. A formative study with amateur and professional designers informed the final interface and provided initial evidence about the approach’s usability and support for exploration. A technical evaluation with 100 stock images characterized processing time, extraction success, output diversity, and failure cases. An expert review and five application scenarios further identified promising uses and the precision and integration improvements needed for professional workflows.

6.5 Frontier Synthesis

SonifyAR and DepthScape demonstrate that a productive human-AI spatial representation must satisfy two audiences. It must be structured enough for computation while remaining perceptible and actionable for a person. SonifyAR’s event textualization gives models a compact description of actors, actions, virtual

objects, and physical context; sound then lets users assess the result through the scene's auditory character. DepthScape's visual programs connect semantic object references to explicit geometry; parametric anchors then let users assess and correct that geometry through direct manipulation.

The frontier advances the dissertation by treating representation and interaction as part of spatial intelligence itself. Human-AI symbiotic spatial perception depends not only on what the AI can infer, but on whether people can perceive the inference, connect it to the scene and their current intent, and shape it through meaningful action. Across sound authoring and 2.5D design, object-centered intermediate representations provide that shared surface for communication.

Chapter 7

Synthesis: Toward Human-AI Symbiotic Spatial Perception

The four frontiers in this dissertation describe different conditions for a future in which people and AI systems can perceive the physical world together. Frontier I considers when assistance is useful: it must be in context, timely, and responsive to the person acting in a scene. Frontier II considers where and when evidence can be acquired: sensing and representation can extend spatial knowledge beyond immediate presence. Frontier III considers whose action an AI judgment supports: spatial claims must remain valid for a particular body, capability, goal, and level of risk. Frontier IV considers how people and AI exchange spatial knowledge: representations must be understandable, expressive, and open to revision. Together, these conditions provide a design account of human-AI symbiotic spatial perception grounded in the systems and evaluations developed throughout the dissertation.

7.1 Contributions

7.1.1 A Four-Frontier Framework

The foundational contribution of this dissertation is a human-centered framework for translating AI world understanding into useful spatial assistance. The framework does not assume that a sufficiently capable model will automatically produce a successful human-AI system. Instead, it identifies four requirements

that connect model capability to physical activity: assistance must be timely and grounded in the current context; evidence must be acquired from the places, times, and viewpoints relevant to the task; spatial judgments must be aligned with the capabilities and goals of the person or agent who will act; and spatial data must be represented through interactions that people can understand and shape. The framework gives the term *human-AI symbiotic spatial perception* an operational meaning and provides a common vocabulary for comparing systems that otherwise span accessibility, mapping, navigation, augmented reality, and visual design.

7.1.2 System, Interaction, and Benchmark Contributions

Frontier I contributes a complete loop for situated accessibility assessment. RASSAR combines mobile LiDAR reconstruction, real-time object detection, machine-readable accessibility rubrics, scan guidance, in-place visual and audio feedback, and post-scan review. Its contribution is not simply a detector. It shows how sensing, interpretation, personalization, and correction can be assembled into an interaction in which a person gathers evidence and reviews AI-generated findings while still in the relevant environment. The stakeholder studies and technical evaluation also identify where this loop remains incomplete, including limited coverage, mismatches between general standards and individual needs, and inaccessible portions of the review experience.

Frontier II contributes complementary methods for extending spatial capture. RAIS integrates robotic navigation with RASSAR’s accessibility-scanning pipeline to test whether predefined audits can be repeated without a person carrying the sensor. FlyMeThrough addresses larger and less structured indoor mapping through commodity-drone video, reconstruction, sparse click-based annotation, segmentation propagation, depth-guided raycasting, and interactive 3D review. The projects establish a continuum of human-AI labor: automation can execute a structured task when the required evidence is known in advance, while human guidance remains necessary when local stakeholders must determine which facilities and objects are meaningful.

Frontier III contributes an evaluation instrument for spatial alignment. CapNav represents indoor navigation through tour videos, semantic graphs, capability profiles, edge-level traversability labels, and metrics for feasibility, path validity, route traversability, and reasoning validity. This decomposition prevents

a plausible final answer from concealing an invalid path or a capability violation. Across 45 scenes, 2,365 navigation tasks, and 5,075 traversability labels, the benchmark provides reusable data and annotation tools while showing that current vision-language models are substantially less reliable when navigation depends on multi-view integration, metric clearance, turning space, or embodiment-specific operation constraints.

Frontier IV contributes interaction techniques for making inferred spatial structure usable in authoring. SonifyAR represents interactions among users, virtual objects, and physical surfaces as contextual events that can guide sound retrieval and generation; programming by demonstration lets creators specify those events through situated action. DepthScape combines monocular depth reconstruction, depth-based blending, visual programs, geometry extraction, and parametric anchors that can be directly manipulated in a 2D viewport. Across the two systems, object- and event-centered intermediate representations connect machine inference to concepts that creators can inspect, preview, accept, and revise.

7.1.3 Evidence About Designing with Imperfect AI

Across all four frontiers, the empirical contribution is an account of what makes imperfect AI useful without treating imperfection as solved. The studies show that people can benefit from automated measurements, propagated annotations, generated sounds, inferred geometry, and model-generated routes, but they also repeatedly expose the boundary of those capabilities. Detection errors require in-place verification; reconstruction gaps limit remote maps; capability-conditioned routing fails on dimensions that are difficult to infer visually; generated sounds vary in quality; and automatically extracted anchors can be geometrically valid yet unsuitable for a design. These findings support the dissertation’s thesis in a bounded form: useful spatial assistance can emerge before AI world understanding is complete when systems constrain the task, expose intermediate results, preserve human control, and evaluate the consequences of errors rather than only aggregate model performance.

7.2 Cross-Frontier Lessons

Across the four frontiers and six projects, six recurring lessons connect the individual contributions. The first three concern how people and AI divide and inspect spatial work. The latter three concern representations of people, stewardship of captured data, and the technical conditions required for deployment. They are

related, but each identifies a distinct design responsibility.

Start with task-relevant evidence. Spatial systems should not treat comprehensive capture as equivalent to useful understanding. RASSAR and RAIS begin with accessibility and safety criteria; FlyMeThrough lets people indicate which facilities a map should retain; CapNav makes capability constraints explicit; and SonifyAR and DepthScape organize generation around meaningful events, objects, and relations. Across these settings, rubrics, routes, surfaces, objects, and events provide compact structures through which people can specify what matters and AI can gather, localize, evaluate, or transform evidence.

Make machine interpretations perceptible and repairable. A geometric reconstruction, model judgment, or generated output becomes useful only when a person can connect it to the observed environment and current task and correct it when necessary. AR markers, 3D bounding boxes, sound previews, navigation graphs, and direct-manipulation handles expose different parts of a system's interpretation. Their value is not merely presentational: they create opportunities to verify a claim, request more evidence, revise an anchor, or reject an inappropriate result. Human-AI collaboration is therefore sustained through feedback and repair rather than unconditional trust [Amershi et al., 2019; Lee and See, 2004].

Use automation to redistribute work while preserving consequential judgment. The projects demonstrate several divisions of labor. A person can provide goals, local knowledge, capability constraints, and evaluative judgment while a system measures geometry, propagates annotations, reconstructs a scene, generates media, or searches for possible routes. A robot or drone can reduce the burden of capture, and an interactive representation can reduce the burden of expressing spatial intent. In each case, automation is most valuable when it makes human attention more selective and effective while preserving authority over the decisions that affect access, safety, navigation, or creative intent.

Representing people is an ongoing design problem. The projects encode people and capabilities in deliberately different ways. RASSAR begins with stakeholder categories and machine-readable rubrics, making the criteria visible and customizable; CapNav uses structured JSON profiles, making dimensions and operational abilities explicit enough to test. A learned or embedded profile could adapt from behavior and context, but it could also make consequential assumptions harder to inspect. Every representation compresses a person's changing body, tools, strategies, preferences, and situation. As one RASSAR participant observed, accessibility standards are "just a fixed standard for reference" and "cannot ensure fitting

on everyone.” Future representations should therefore be *living*, so they can change with current conditions; *detailed*, so they can express interacting and nuanced capabilities; *transparent*, so people can understand why a profile produced a result; *editable*, so people can correct it; and *verifiable*, so its predictions can be checked against real observations and actions.

Data management determines whether captured space becomes usable. Phones, cameras, robots, drones, and building systems already make widespread spatial capture technically plausible. The more difficult problem often begins after collection: deciding what to retain, filtering sensitive information, recording provenance and freshness, controlling who can inspect or reuse each representation, and compressing large image, video, and geometry streams into task-relevant forms. Edge processing can reduce transmission and expose compact derived information instead of raw capture, but it does not by itself settle ownership, consent, or future access. The benefit of spatial data may therefore depend less on whether it can be collected than on whether it can be protected, maintained, and made appropriately accessible.

Model performance remains a deployment bottleneck. Across the projects, models that perform reasonably in controlled prototypes still make errors that prevent routine reliance in consequential spatial tasks. Deployment requires improvement not only in average accuracy, but also in robustness to changing viewpoints and environments, calibration under missing evidence, and reliability for capability-specific cases. Latency compounds this problem: continuous guidance may require subsecond responses, while current multimodal pipelines can take many seconds or longer on complex inputs. More efficient models, local or edge execution, and task-specific combinations of learned perception with geometry and classical planning are therefore necessary complements to interface-level safeguards. Section 7.4 examines these constraints in greater detail.

7.3 Users, Stakeholders, and Scope of Evidence

The word *user* refers to several different roles across this dissertation. A person may be the intended beneficiary whose capabilities and goals define a system requirement, the operator who captures or annotates spatial data, a domain stakeholder who contributes local or professional knowledge, or a study participant whose behavior and feedback support an empirical claim. These roles sometimes coincide.

RASSAR most directly centers disability and accessibility stakeholders. Its formative work involved

wheelchair users, blind and low-vision people, caregivers, families with young children, and occupational therapists; its later evaluation again included people with lived or professional accessibility experience. These participants shaped the rubrics, scan guidance, feedback modalities, and emphasis on customization. RAIS inherits that accessibility task definition, but its evaluation is a preliminary technical study rather than a study with disabled end users. FlyMeThrough addresses a different research question: whether people with site-specific knowledge can guide the creation of indoor semantic maps. It was therefore evaluated with building managers and active occupants, who could judge facilities and mapping workflows but were not recruited to represent disabled visitors or accessibility expertise.

CapNav evaluates representations of capability rather than the lived navigation experience of those represented. Its five profiles include two human cases, an adult without motor disabilities and a wheelchair user, along with three robotic embodiments. The wheelchair profile makes capability-conditioned failures measurable, but its constraints and ground-truth annotations were produced by the research team rather than validated by wheelchair users navigating the spaces. The benchmark therefore supports claims about model sensitivity to specified constraints, not claims that its profile captures the diversity of wheelchair use or that benchmark success guarantees accessible navigation. Similarly, SonifyAR discusses the accessibility potential of sound, but its eight-person study evaluated AR authoring and did not recruit blind or low-vision creators. DepthScape was evaluated with design enthusiasts and professional designers because its immediate contribution concerns 2.5D visual authoring.

These populations were selected to match each project’s immediate research question and stage of development: lived and professional accessibility knowledge for RASSAR, local building knowledge for FlyMeThrough, controlled annotation for a diagnostic benchmark, and relevant creative practice for the authoring tools. That rationale does not remove the resulting limitations. Evidence that a workflow is technically feasible, understandable to a proxy operator, or useful to a creator cannot be generalized into evidence of benefit for a disability community. Disability is also not a single capability profile; needs vary with body, sensory access, mobility technology, environment, task, experience, and intersecting identities.

Future work should align participation with the claims being made at each stage. Systems that claim accessibility benefits should involve the relevant disabled communities in problem formulation, design, and evaluation, including longitudinal use in consequential settings. Capability profiles should be treated as

editable hypotheses and compared against lived strategies rather than as complete descriptions of a group. Studies should also examine who performs the labor of scanning, annotation, correction, and maintenance, who controls the resulting spatial data, and whose environment is treated as the default. This more explicit account of participation would strengthen both the validity and the accountability of human-AI symbiotic spatial perception.

7.4 Current Deployment Constraints

By designing, implementing, deploying, and evaluating projects across the four frontiers, this dissertation also makes clear why these systems are not yet everyday spatial infrastructure.

Timeliness, accuracy, and bounded context. Spatial assistance depends on an end-to-end pipeline: sensing, inference, network transmission when models are remote, and feedback must all occur before a decision has passed. The relevant latency target is determined by the action being supported, not simply by whether a system feels responsive. In a classic target-acquisition study, performance costs were already measurable at 75 ms of system lag; at 225 ms, movement time rose by 63.9%, error rate rose by 214%, and throughput fell by 46.5% relative to an 8.3 ms condition [MacKenzie and Ware, 1993]. These values are not a universal deadline for every spatial task, but they make the central point vivid: when advice is meant to guide an ongoing action, an answer that arrives after the action has moved on is less useful, even if it is correct.

Under the task-relative definition introduced in Chapter 1, the end-to-end workflows in this dissertation span seconds to hours. RASSAR provides incremental feedback during a scan, but its full researcher-led scans averaged 99.9 seconds and participant-led scans averaged 3.3 minutes. FlyMeThrough first requires a drone flight and then a reconstruction pipeline that can take several hours per scene, although later annotation stages can be selectively applied to the resulting model. CapNav’s mean non-thinking inference time was 14.94 seconds, and enabling its thinking setting increased that mean to 123.94 seconds. These timing profiles can support inspection, planning, and deliberative decision making, but they do not satisfy the subsecond latency needed for continuous locomotion guidance or collision avoidance. A workflow can therefore provide real-time incremental feedback without completing every stage in real time; latency must be evaluated against the decision horizon of each supported action.

Faster models make some local functions more plausible, but only by narrowing the task, model capacity, input resolution, or all three. A recent paper on micro-NPU vision-language inference reports a 117.8 CIDEr score for image captioning under a 32 MB constraint and describes millisecond-level inference; this is encouraging evidence for specialized local VLM components, not an apples-to-apples solution to multi-view accessibility assessment or navigation reasoning.¹ More broadly, efficient-VLM research reports both high throughput on mobile and edge hardware and recurring accuracy-efficiency tradeoffs: for example, MobileVLM reached 21.5 tokens/s on a Snapdragon 888 CPU and 65.3 tokens/s on a Jetson Orin GPU, whereas an efficiency-oriented system retained 98.4% of its reference performance while obtaining a $2.2\times$ speedup [Shinde et al., 2025]. Richer context can also increase latency [Chang et al., 2024], while long tours and evolving scenes require models to remember, select, and integrate observations across time. Current multimodal models remain unreliable on multi-view and metric spatial judgments [Yang et al., 2025b] and generalize unevenly from structured indoor settings to open-world environments [Wu et al., 2026]. Object and relation hallucinations further mean that a fluent answer can be unsupported by the available evidence [Li et al., 2023; Guan et al., 2024].

Cloud deployment adds a network-bound delay and makes its tail behavior part of the interaction design. A recent study of remote visual inference treats round-trip delays from 0 to 165 ms as a realistic operating range and uses 100 ms as a concrete evaluation point; its central finding is that remote predictions become stale as the world changes, motivating a fresh local fallback rather than reliance on the remote result alone [Jacobellis et al., 2025]. Consequently, future systems must evaluate the full pipeline, including capture, encoding, transmission, queueing, inference, and feedback, and choose an explicit balance among timing, accuracy, cost, energy, and safety for the task at hand.

Devices and physical capture. Phone LiDAR, cameras, robotic platforms, drones, and head-mounted displays make new forms of spatial assistance possible, but their availability and reliability are uneven. Each constrains sensing range, viewpoint, battery life, computational budget, ergonomics, connectivity, and safe operation. A phone may be widely available but miss occluded features, surfaces outside its field of view, or small and movable hazards. A robot or drone can gather additional viewpoints, but requires permission, time,

¹Z. Chen et al., “muVLM: A Vision Language Model for muNPUs,” CVPR 2026. Open-access version: https://openaccess.thecvf.com/content/CVPR2026/papers/Chen_muVLM_A_Vision_Language_Model_for_mNPUs_CVPR_2026_paper.pdf.

maintenance, safe operation, and access to the place being captured. Even within a successful capture, visual similarity, transparent or reflective materials, repeated geometry, dynamic objects, and changing layouts can make reconstruction or semantic interpretation incomplete.

It is therefore unrealistic to expect the physical world to become comprehensively scanned and continuously current. Spatial AI will operate with mixed levels of world knowledge: some places will have dense, recent, task-specific capture; others will have sparse or stale observations; and many will have no usable digital evidence at all. A system should treat this heterogeneity as an explicit part of its representation rather than silently filling every gap with a confident world model. It should distinguish what was directly observed from what was reconstructed, inferred, predicted, or imported from another source; communicate the spatial and temporal extent of its evidence; and identify when an answer depends on missing, old, or weakly supported data. This provenance and uncertainty are especially important when a judgment affects mobility, safety, or access.

The resulting design problem is not simply to maximize capture quality. It is to balance quality, availability, burden, and timeliness for a particular task. A partial but recent scan may be sufficient for an immediate local question, while a building-management decision may require corroborating measurements, review, and a renewed capture.

Privacy, consent, and governance. Spatial capture can reveal layout, object existence, activity patterns, and personal habits. These data become especially sensitive when images or scans leave the device for cloud processing, when sensing is repeated over time, or when a drone or robot enters a shared space. The projects in this dissertation generally reduce exposure through local or tightly controlled capture and processing, but local deployment alone does not resolve bystander exposure, secondary use, re-identification, or the question of who may inspect an enduring spatial model.

Recent mixed-reality privacy research frames these risks as a design problem involving both the person operating a device and people who happen to share the space. Work on bystander privacy emphasizes protection that does not make the primary interaction unusable [Corbett et al., 2024]. Research on spontaneous AR interactions develops consent mechanisms for contexts in which people have not had an opportunity to negotiate capture in advance [Windl et al., 2025], while work on privacy indicators argues that cues must remain perceivable even when a bystander is situationally impaired [Bukhari et al., 2025]. Together, these

contributions move beyond a one-time privacy notice toward privacy mechanisms that are visible, contextual, and usable by more than the system’s owner.

Future spatial systems should therefore collect the minimum evidence needed for the present task, prefer local processing and transformed data when feasible, and make capture states legible through persistent and multimodal cues. They should support granular, context-specific, and revocable consent; set bounded retention and sharing rules for raw imagery, reconstructions, and derived annotations; and provide people with a meaningful way to inspect, correct, remove, or contest data about spaces they inhabit. When a cloud service is necessary, the system should disclose what leaves the device, which party can access it, how long it is retained, and how the response will degrade if permission or connectivity is unavailable. Privacy cannot be reduced to a notice shown after capture: it must shape what is sensed, retained, transmitted, shared, and made visible to different people.

7.5 Future Directions

Although this dissertation concludes my PhD, I remain committed to advancing the vision of human-AI symbiotic spatial perception through future research and engineering. Three directions are especially important:

Edge-native and privacy-preserving spatial inference. Future systems should move more sensing and inference onto local devices whenever doing so improves responsiveness and keeps sensitive observations under the user’s control. This does not require that every model run entirely on-device. It requires systems that make a deliberate, inspectable allocation between local and remote computation, minimize transmitted evidence, preserve useful uncertainty when the local model cannot answer a question, and retain a safe local fallback when remote results are delayed or unavailable [Jacobellis et al., 2025]. Such an approach can reduce network-dependent delays while better protecting both users and bystanders.

Spatial interaction beyond the screen. Shared spatial representations should be available through the modalities in which people already understand and act in the world: direct manipulation, sound, embodied motion, physical reference, and situated AR feedback. Recent HCI work shows how direct manipulation can complement language when people need to communicate complex visual intent [Chen et al., 2025], and current reviews identify a broadening design space for interaction beyond text-only prompting [Park

et al., 2026]. The next challenge is to carry these ideas into everyday physical environments, where a person should be able to point, move, demonstrate, listen, and correct an AI system without translating every spatial relationship into a digital command.

Living, editable models of people and context. The dissertation uses textual rubrics and capability profiles because they make assumptions visible and testable. They are nevertheless limited snapshots of a person's abilities, goals, preferences, environment, and changing circumstances. Future systems need richer representations that can be updated from context and interaction while remaining transparent, editable, and under the person's control. Personalized accessibility scanning and shared-control navigation provide early examples of systems that adapt to individual mobility, preferences, and desired levels of involvement [Huang et al., 2025; Kamikubo et al., 2025]. A mature user model should preserve this agency: it must show what the system currently believes, let people change it, and avoid silently turning a temporary condition into a permanent identity.

Human-AI symbiotic spatial perception will emerge not from a single universal world model, but from systems that can form, communicate, and revise task-specific shared representations. The research agenda ahead is to make those representations faster, more trustworthy, more private, and more responsive to the person whose world they describe.

Chapter 8

Conclusion

This dissertation began with a simple question: how can AI extend human spatial perception without displacing human judgment? The answer developed here is not a single system or sensing technology. It is a form of partnership in which AI contributes evidence, inference, and computational reach while people retain the ability to interpret, question, and redirect the spatial representations on which action depends.

Across four frontiers, the dissertation explored the conditions for that partnership. RASSAR examined situated, personalized assistance; RAIS and FlyMeThrough extended the reach of spatial evidence; CapNav made capability alignment a measurable requirement for spatial reasoning; and SonifyAR and DepthScape treated representation and interaction as core elements of spatial intelligence. Together, these projects show that useful spatial AI must be timely enough to matter, expansive enough to reach beyond immediate presence, aligned with the person who will act, and expressive enough for people to inspect and revise.

Human-AI symbiotic spatial perception is therefore not the replacement of physical experience with automated descriptions. It is the careful extension of what people can know and do in the physical world. Its success will be measured by whether an AI system helps someone act with better evidence, clearer understanding, and greater agency in the places that matter to them.

Bibliography

ARVid - Augmented Reality. Accessed on September 24, 2023.

Freesound. Accessed on September 24, 2023.

Halo AR. Accessed on September 24, 2023.

Department of Justice . 2010. ADA Standards for Accessible Design.

Adobe. 2023. Adobe audition sound effects download page. Accessed on Date of Access.

Adobe. Accessed September 11, 2023. Adobe aero. <https://www.adobe.com/products/aero.html>.

Dragan Ahmetovic, Cole Gleason, Chengxiong Ruan, Kris M. Kitani, Hironobu Takagi, and Chieko Asakawa. 2016. Navcog: A navigational cognitive assistant for the blind. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (Mobile-HCI)*, pages 90–99.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do as i can, not as i say: Grounding language in robotic affordances.

- Meta AI. 2024. Segment anything model 2 (sam2). <https://github.com/facebookresearch/sam2>. Accessed: 2025-04-06.
- Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI)*. ACM.
- Peter Anderson and et al. 2018a. On evaluation of embodied navigation agents. In *arXiv:1807.06757*.
- Peter Anderson and et al. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*.
- Aniwaa. 2020. Faro focus swift. Accessed: 2024-08-15.
- Apple. Core ML.
- Apple. 2020. Apple introduces iPhone 12 Pro and iPhone 12 Pro Max with 5G.
- Apple. 2022. RoomPlan - Augmented Reality.
- Apple. 2023a. Apple arkit documentation. Accessed on Oct 9th, 2023.
- Apple. 2023b. Arkit - tracking and visualizing planes. Accessed on Oct 9th, 2023.
- Apple. 2023c. Raycast.
- Apple. 2023d. Scenekit - physics simulation. Accessed on Oct 9th, 2023.
- Apple. Accessed September 11, 2023. Reality composer. <https://apps.apple.com/us/app/reality-composer/id1462358802>.
- V. Ayala-Alfaro, J. A. Vilchis-Mar, F. E. Correa-Tome, and J. P. Ramirez-Paredes. 2021. Automatic Mapping with Obstacle Identification for Indoor Human Mobility Assessment. *ArXiv:2111.12690 [cs]*.
- Ruzena Bajcsy. 1988. Active perception. *Proceedings of the IEEE*, 76(8):996–1005.

- J. Balado, L. Díaz-Vilariño, P. Arias, and M. Soilán. 2017. Automatic building accessibility diagnosis from point clouds. *Automation in Construction*, 82:103–111.
- Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. 2025. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15791–15801.
- Shmuel Berman and Jia Deng. 2025. Vlms have tunnel vision: Evaluating nonlocal visual reasoning in leading vlms. *arXiv preprint arXiv:2507.13361*.
- Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. 2024. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*.
- Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, 11(4):589–597. Publisher: Routledge _eprint: <https://doi.org/10.1080/2159676X.2019.1628806>.
- Eli Brenner and Jeroen B Smeets. 2018. Depth perception. In *Stevens’ handbook of experimental psychology and cognitive neuroscience: Sensation, perception, and attention*, pages 385–414. Wiley.
- Syed Ibrahim Mustafa Shah Bukhari, Maha Sajid, Bo Ji, and Brendan David-John. 2025. Rethinking privacy indicators in extended reality: Multimodal design for situationally impaired bystanders. In *2025 IEEE International Symposium on Mixed and Augmented Reality Adjunct*, pages 265–272.
- Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John J. Leonard. 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332.
- Shaojun Cai, Ashwin Ram, Zhengtai Gou, Mohd Alqama Wasim Shaikh, Yu-An Chen, Yingjia Wan, Kotaro Hara, Shengdong Zhao, and David Hsu. 2024. Navigating real-world challenges: A quadruped robot guiding system for visually impaired people in diverse environments. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI ’24, New York, NY, USA. Association for Computing Machinery.

- Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision*, pages 667–676.
- Ruei-Che Chang, Yuxuan Liu, and Anhong Guo. 2024. Worldscribe: Towards context-aware live visual descriptions. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–18.
- Kathy Charmaz. 2006. *Constructing grounded theory: A practical guide through qualitative analysis*. sage.
- Juntong Chen, Jiang Wu, Jiajing Guo, Vikram Mohanty, Xueming Li, Jorge Piazzentin Ono, Wenbin He, Liu Ren, and Dongyu Liu. 2025. Interchat: Enhancing generative visual analytics using multimodal interactions. *Computer Graphics Forum*, 44(3):e70112.
- An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. 2024. Spatialrgpt: Grounded spatial reasoning in vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Neil Chulpongsatorn, Mille Skovhus Lunding, Nishan Soni, and Ryo Suzuki. 2023. Augmented math: Authoring ar-based explorable explanations by augmenting static math textbooks. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA. Association for Computing Machinery.
- Gregory D. Clemenson, Antonella Maselli, Alexander J. Fiannaca, Amos Miller, and Mar Gonzalez-Franco. 2021. Rethinking GPS navigation: Creating cognitive maps through auditory clues. *Scientific Reports*, 11:7764.
- Lindy Clemson. 1997. *Home Fall Hazards: A Guide to Identifying Fall Hazards in the Homes of Elderly People and an Accompaniment to the Assessment Tool, The Westmead Home Safety Assessment (WeHSA)*. Co-ordinates Publications.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with

- advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Matthew L. Corbett, Brendan David-John, Jiacheng Shang, Y. Charlie Hu, and Bo Ji. 2024. Securing bystander privacy in mixed reality while protecting the user experience. *IEEE Security & Privacy*, 22(1):33–42.
- James M Coughlan and Joshua Miele. 2017. Ar4vi: Ar as an accessibility tool for people with visual impairments. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 288–292. IEEE.
- Abhraneil Dam, Arsh Siddiqui, Charles Leclercq, and Myounghoon Jeon. 2024. Taxonomy and definition of audio augmented reality (aar): A grounded theory study. *International Journal of Human-Computer Studies*, 182:103179.
- Thi Hong Diep Dao and Jean-Claude Thill. 2018. Three-dimensional indoor network accessibility auditing for floor plan design. *Transactions in GIS*, 22(1):288–310. [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/tgis.12310](https://onlinelibrary.wiley.com/doi/pdf/10.1111/tgis.12310).
- Hui Deng, Mao Tian, Zhibin Ou, and Yichuan Deng. 2022. A semantic framework for on-site evacuation routing based on awareness of obstacle accessibility. *Automation in Construction*, 136:104154.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- Darko Dimitrov, Christian Knauer, Klaus Kriegel, and Günter Rote. 2006. On the bounding boxes obtained by principal component analysis. In *22nd European Workshop on Computational Geometry*, pages 193–196.
- Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff,

- Andy Zeng, Igor Mordatch, and Pete Florence. 2023. PaLM-E: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 8469–8488. PMLR.
- Lucia Díaz Vilariño, Ha Tran, Ernesto Frías, Jesus Balado Frias, and Kourosh Khoshelham. 2022. 3D MAPPING OF INDOOR AND OUTDOOR ENVIRONMENTS USING APPLE SMART DEVICES. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2022:303–308.
- Lucía Díaz-Vilariño, Pawel Boguslawski, Kourosh Khoshelham, and Henrique Lorenzo. 2019. Obstacle-Aware Indoor Pathfinding Using Point Clouds. *ISPRS International Journal of Geo-Information*, 8(5):233. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.
- Basem M. ElHalawany, Hala M. Abdel-Kader, Adly TagEldeen, Alaa Eldeen Elsayed, and Zaki B. Nossair. 2013. Modified A* algorithm for safer mobile robot navigation. In *2013 5th International Conference on Modelling, Identification and Control (ICMIC)*, pages 74–78.
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.
- Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. 2025. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*.
- Mohd Ihsan Alimi Mohd Filus and Dayang Rohaya Awang Rambli. 2012. Using non-speech sound as acoustic modality in augmented reality environment. In *2012 International Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, pages 79–82. IEEE.
- Flyability. 2025. Elios 3: The ultimate indoor drone for inspection and mapping. Accessed: 2025-02-14.
- Jon Froehlich, Leah Findlater, Marilyn Ostergren, Solai Ramanathan, Josh Peterson, Inness Wragg, Eric Larson, Fabia Fu, Mazhengmin Bai, Shwetak Patel, et al. 2012. The design and evaluation of prototype eco-feedback displays for fixture-level water usage data. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2367–2376.

- Jon E. Froehlich, Anke M. Brock, Anat Caspi, João Guerreiro, Kotaro Hara, Reuben Kirkham, Johannes Schöning, and Benjamin Tannert. 2019. Grand challenges in accessible maps. *Interactions*, 26(2):78–81.
- Qiang Fu, Hongbo Fu, Hai Yan, Bin Zhou, Xiaowu Chen, and Xueming Li. 2020. Human-centric metrics for indoor scene assessment and synthesis. *Graphical Models*, 110:101073.
- Agneta Fänge and Susanne Iwarsson. 2005. Changes in accessibility and usability in housing: an exploration of the housing adaptation process. *Occupational Therapy International*, 12(1):44–59. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/oti.14>.
- Chuanxiang Gao, Xinyi Wang, Xi Chen, and Ben M Chen. 2024. A hierarchical multi-uav cooperative framework for infrastructure inspection and reconstruction. *Control Theory and Technology*, 22(3):394–405.
- Chuanxiang Gao, Xinyi Wang, Ruoyu Wang, Zuoquan Zhao, Yu Zhai, Xi Chen, and Ben M Chen. 2023. A uav-based explore-then-exploit system for autonomous indoor facility inspection and scene reconstruction. *Automation in Construction*, 148:104753.
- Sanchita Ghose and John J Prevost. 2022. Foleygan: Visually guided generative adversarial network-based synchronous sound generation in silent videos. *IEEE Transactions on Multimedia*.
- James J. Gibson. 1979. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, MA.
- Nicholas A. Giudice. 2018. Navigating without vision: Principles of blind spatial cognition. In Daniel R. Montello, editor, *Handbook of Behavioral and Cognitive Geography*, pages 260–288. Edward Elgar Publishing, Cheltenham, UK.
- Nicholas A. Giudice, Jonathan Z. Bakdash, and Gordon E. Legge. 2007. Wayfinding with words: Spatial learning and navigation using dynamically updated verbal descriptions. *Psychological Research*, 71(3):347–358.
- Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. 2024. End-to-end navigation with vision language models: Transforming spatial reasoning into question-answering. *arXiv preprint arXiv:2411.05755*.

- Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2024. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385.
- Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962.
- David Ha and Jürgen Schmidhuber. 2018. World models. *arXiv preprint arXiv:1803.10122*.
- Torsten Hägerstrand. 1970. What about people in regional science? *Papers in Regional Science*, 24(1):7–24.
- Kotaro Hara, Christine Chan, and Jon E Froehlich. 2016. The design of assistive location-based technologies for people with ambulatory disabilities: A formative study. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 1757–1768.
- Kotaro Hara, Vicki Le, and Jon Froehlich. 2013. Combining crowdsourcing and google street view to identify street-level accessibility problems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 631–640, Paris France. ACM.
- Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.
- Beverly P. Horowitz, Tiffany Almonte, and Andrea Vasil. 2016. Use of the Home Safety Self-Assessment Tool (HSSAT) within Community Health Education to Improve Home Safety. *Occupational Therapy In Health Care*, 30(4):356–372. Publisher: Taylor & Francis _eprint: <https://doi.org/10.1080/07380577.2016.1191695>.
- Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 159–166.

- United States Department of Housing and Urban Development Office of Housing and Barrier Free Environments inc. 1996. *Fair Housing Act Design Manual: A Manual to Assist Designers and Builders in Meeting the Accessibility Requirements of the Fair Housing Act*. U.S. Department of Housing and Urban Development, Office of Fair Housing and Equal Opportunity and the Office of Housing.
- Ian P Howard and BJ Rogers. 2002. Depth perception. *Stevens' handbook of experimental psychology Sensation and Perception*, 1:77–120.
- Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. 2022. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*.
- William Huang, Xia Su, Jon E. Froehlich, and Yang Zhang. 2025. Accessibility scout: Personalized accessibility scans of built environments. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pages 1–18.
- John Hughes. 2021. krippendorffsalpha: An r package for measuring agreement using krippendorff's alpha coefficient. *arXiv preprint arXiv:2103.12170*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Gabriel Ilharco and et al. 2019. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv:1907.05446*.
- Rob Imrie. 2003. Housing quality and the provision of accessible homes. *Housing Studies*, 18(3):387–408.
- EyeSpy360 Inc. 2024a. Eyespy360: 360 virtual tour software. <https://www.eyespy360.com/en-us/>. Accessed: 2024-08-15.
- HP Inc. 2024b. Hp siteprint - robotic layout solution. Accessed: 2024-08-15.
- Matterport Inc. 2024c. Matterport: 3d capture platform. <https://matterport.com/>. Accessed: 2024-08-15.

- Metareal Inc. 2024d. Metareal: Virtual tours made simple. <https://www.metareal.com/>. Accessed: 2024-08-15.
- Ferli Septi Irwansyah, YM Yusuf, Ida Farida, and Muhammad Ali Ramdhani. 2018. Augmented reality (ar) technology on the android operating system in chemistry learning. In *IOP conference series: Materials science and engineering*, volume 288, page 012068. IOP Publishing.
- Susanne Iwarsson. 1999. The Housing Enabler: An Objective Tool for Assessing Accessibility. *British Journal of Occupational Therapy*, 62(11):491–497. Publisher: SAGE Publications Ltd STM.
- Dan Jacobellis, Mateen Ulhaq, Fabien Racapé, Hyomin Choi, and Neeraja J. Yadwadkar. 2025. Dedelayed: Deleting remote inference delay via on-device correction. *arXiv preprint arXiv:2510.13714*.
- Dhruv Jain, Leah Findlater, Jamie Gilkeson, Benjamin Holland, Ramani Duraiswami, Dmitry Zotkin, Christian Vogler, and Jon E. Froehlich. 2015. Head-Mounted Display Visualizations to Support Sound Awareness for the Deaf and Hard of Hearing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 241–250, New York, NY, USA. Association for Computing Machinery.
- Dhruv Jain, Sasa Junuzovic, Eyal Ofek, Mike Sinclair, John R. Porter, Chris Yoon, Swetha Machanavajhala, and Meredith Ringel Morris. 2021. A Taxonomy of Sounds in Virtual Reality. In *Proceedings of the 2021 International Conference on Multimodal Interaction*, pages 80–91, Montréal QC Canada. ACM.
- Gaurav Jain, Leah Findlater, and Cole Gleason. 2026. Scenescout: Towards ai-driven access to street level imagery for blind users. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–22.
- Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. *arXiv preprint arXiv:1905.12255*.
- Won Shik Jang and Ue-Hwan Kim. 2026. Context-nav: Context-driven exploration and viewpoint-aware 3d spatial reasoning for instance navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9626–9636.

- Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, Zeng Yifu, Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. 2022. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.
- Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3668–3678.
- U.S. Department of Justice. 2024. 2010 ada standards for accessible design. <https://www.ada.gov/law-and-regs/design-standards/>. Accessed: 2024-08-15.
- Rie Kamikubo, Seita Kayukawa, Yuka Kaniwa, Allan Wang, Hernisa Kacorri, Hironobu Takagi, and Chieko Asakawa. 2025. Beyond omakase: Designing shared control for navigation robots with blind people. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–17.
- Seokbin Kang, Leyla Norooz, Elizabeth Bonsignore, Virginia Byrne, Tamara Clegg, and Jon E. Froehlich. 2019. Prototypar: Prototyping and simulating complex systems with paper craft and augmented reality. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children, IDC '19*, page 253–266, New York, NY, USA. Association for Computing Machinery.
- Seokbin Kang, Leyla Norooz, Vanessa Oguamanam, Angelisa C. Plane, Tamara L. Clegg, and Jon E. Froehlich. 2016. Sharedphys: Live physiological sensing, whole-body interaction, and large-screen visualizations to support shared inquiry experiences. In *Proceedings of the The 15th International Conference on Interaction Design and Children, IDC '16*, page 275–287, New York, NY, USA. Association for Computing Machinery.
- Seokbin Kang, Ekta Shokeen, Virginia L. Byrne, Leyla Norooz, Elizabeth Bonsignore, Caro Williams-Pierce, and Jon E. Froehlich. 2020. Armath: Augmenting everyday life with math learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20*, page 1–15, New York, NY, USA. Association for Computing Machinery.

- Samer Karam, Francesco Nex, Bhanu Teja Chidura, and Norman Kerle. 2022. Microdrone-based indoor mapping with graph slam. *Drones*, 6(11):352.
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1.
- Kids Clinic. 2023. Home safety checklist. [Online; accessed 8-December-2023].
- Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. 2021. Pathdreamer: A world model for indoor navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14738–14748.
- Nemanja Kostic and Simon Scheider. 2015. Automated Generation of Indoor Accessibility Information for Mobility-Impaired Individuals. In Fernando Bacao, Maribel Yasmina Santos, and Marco Painho, editors, *AGILE 2015*, pages 235–252. Springer International Publishing, Cham. Series Title: Lecture Notes in Geoinformation and Cartography.
- Jacob Krantz and et al. 2020. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*.
- Klaus Krippendorff. 2011. Computing krippendorff’s alpha-reliability.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25.
- Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Mallocci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. 2020. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*.
- Tessa A. Lau and Daniel S. Weld. 1998. Programming by demonstration: An inductive learning formulation. In *Proceedings of the 4th International Conference on Intelligent User Interfaces*, IUI ’99, page 145–152, New York, NY, USA. Association for Computing Machinery.

- Mary Law, Barbara Cooper, Susan Strong, Debra Stewart, Patricia Rigby, and Lori Letts. 1996. The person-environment-occupation model: A transactive approach to occupational performance. *Canadian journal of occupational therapy*, 63(1):9–23.
- Jaewook Lee, Filippo Aleotti, Diego Mazala, Guillermo Garcia-Hernando, Sara Vicente, Oliver James Johnston, Isabel Kraus-Liang, Jakub Powierza, Donghoon Shin, Jon E. Froehlich, Gabriel Brostow, and Jessica Van Brummelen. 2025. Imaginatear: Ai-assisted in-situ authoring in augmented reality. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pages 1–21.
- John D. Lee and Katrina A. See. 2004. Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80.
- Vincent Leroy, Yohann Cabon, and Jérôme Revaud. 2024. Grounding image matching in 3d with mast3r.
- Fangyu Li, Sisi Zlatanova, Martijn Koopman, Xueying Bai, and Abdoulaye Diakité. 2018. Universal path planning for an indoor drone. *Automation in Construction*, 95:275–283.
- Xiao Li, Bharat Gandhi, Ming Zhan, Mohit Nehra, Zhicheng Zhang, Yuchen Sun, Meijia Song, Naisheng Zhang, and Xi Wang. 2025. Fine-tuning vision-language models for visual navigation assistance. *arXiv preprint arXiv:2509.07488*.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305. Association for Computational Linguistics.
- J. C. R. Licklider. 1960. Man–computer symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1:4–11.
- Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. 2023. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*.
- Tiffany Liu, Javier Hernandez, Mar Gonzalez-Franco, Antonella Maselli, Melanie Kneisel, Adam Glass, Jarnail Chudge, and Amos Miller. 2022. Characterizing and predicting engagement of blind and low-

- vision people with an audio-based navigation app. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–7. Association for Computing Machinery.
- Xiaoan Liu, Difan Jia, Xianhao Carton Liu, Mar Gonzalez-Franco, and Chen Zhu-Tian. 2025. Reality proxy: Fluid interactions with real-world objects in mr via abstract representations. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pages 1–16.
- Richard G. Long and Nicholas A. Giudice. 2010. Establishing and maintaining orientation for mobility. In Bruce B. Blasch, William R. Wiener, and Richard L. Welsh, editors, *Foundations of Orientation and Mobility: History and Theory*, 3 edition, volume 1, pages 45–62. American Foundation for the Blind Press, New York, NY.
- Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, et al. 2019. Mediapipe: A framework for perceiving and processing reality. In *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, volume 2019.
- Kevin Lynch. 1960. *The Image of the City*. MIT Press.
- Mehdi Maboudi, MohammadReza Homaei, Soohwan Song, Shirin Malihi, Mohammad Saadatseresht, and Markus Gerke. 2023. A review on viewpoints and path planning for uav-based 3-d reconstruction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:5026–5048.
- I. Scott MacKenzie and Colin Ware. 1993. Lag as a determinant of human performance in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 488–493.
- Leo Mariga. pyransac-3d. <https://leomariga.github.io/pyRANSAC-3D/>. Accessed: 2025-04-03.
- Microsoft. Bing image search api: Microsoft bing.
- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng.

2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106.
- Daniel R. Montello. 1993. Scale and multiple psychologies of space. In *Spatial Information Theory: A Theoretical Basis for GIS*, volume 716 of *Lecture Notes in Computer Science*, pages 312–321. Springer.
- Raul Mur-Artal and Juan D. Tardós. 2017. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.
- Riku Murai, Eric Dexheimer, and Andrew J. Davison. 2024. MAST3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. *arXiv preprint*.
- John Murray. 1994. Some perspectives on visual depth perception. *ACM SIGGRAPH Computer Graphics*, 28(2):155–157.
- NavVis. 2024. Navvis vlx 3. Accessed: 2024-08-15.
- Thien Nguyen, Elizabeth M Combs, Pamela J Wright, and Cynthia F Corbett. 2021. Reducing fall risks among visually impaired older adults. *Home healthcare now*, 39(4):186–193.
- Shirley Niemeyer and Michael P Vogel. 1994. G94-1213 child care environment safety checklist. *Historical Materials from University of Nebraska-Lincoln Extension*, page 1325.
- Nils J. Nilsson. 1984. Shakey the robot. Technical Report Technical Note 323, SRI International, Artificial Intelligence Center, Menlo Park, CA.
- Donald A. Norman. 1999. Affordance, conventions, and design. *Interactions*, 6(3):38–43.
- Donald A. Norman. 2013. *The Design of Everyday Things*, revised and expanded edition. Basic Books.
- OpenAI. 2024. Gpt-4 technical report.
- OpenAI. 2025a. Introducing gpt-4.1 in the api. <https://openai.com/index/gpt-4-1/>. Accessed: 2025-11-12.
- OpenAI. 2025b. Introducing gpt-5. <https://openai.com/index/introducing-gpt-5/>. Accessed: 2025-11-12.

World Health Organization. 2018. WHO Housing and health guidelines.

Helen Osborne, Tom Scott, and United Spinal Association. 2008. Home Safety for People With Disabilities. *in Motion*, 18(5).

Jin-Si R Over, Andrew C Ritchie, Christine J Kranenburg, Jenna A Brown, Daniel D Buscombe, Tom Noble, Christopher R Sherwood, Jonathan A Warrick, and Phillipe A Wernette. 2021. Processing coastal imagery with agisoft metashape professional edition, version 1.6—structure from motion workflow documentation. Technical report, US Geological Survey.

Hyerim Park, Malin Eiband, Andre Luckow, and Michael Sedlmair. 2026. Interaction methods in generative ai image tools: A review of trends and design opportunities across hci and industry. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–27.

Suraj Patni, Aradhye Agarwal, and Chetan Arora. 2024. Ecodepth: Effective conditioning of diffusion models for monocular depth estimation. *ArXiv*, abs/2403.18807.

Siyu Pei, Alexander Chen, Chen Chen, Franklin Mingzhe Li, Megan Fozzard, Hao-Yun Chi, Nadir Weibel, Patrick Carrington, and Yang Zhang. 2023. Embodied exploration: Facilitating remote accessibility assessment for wheelchair users with virtual reality. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, New York, NY, USA. Association for Computing Machinery.

Pexels. 2025. Pexels – Free Stock Photos and Videos. Accessed: 2025-04-06.

Georgios Pipelidis, Christian Prehofer, and Ilias Gerostathopoulos. 2019. Bootstrapping the dynamic generation of indoor maps with crowdsourced smartphone sensor data. In *Geographical Information Systems Theory, Applications and Management: Third International Conference, GISTAM 2017, Porto, Portugal, April 27–28, 2017, Revised Selected Papers 3*, pages 70–84. Springer.

David MW Powers. 2020. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

- Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. 2025. Spatialvla: Exploring spatial representations for visual-language-action models. In *Proceedings of Robotics: Science and Systems*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastri, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021a. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastri, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021b. Learning transferable visual models from natural language supervision.
- Iulian Radu and Bertrand Schneider. 2019. What can we learn from augmented reality (ar)? benefits and drawbacks of ar for inquiry-based learning of physics. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12.
- Ismo Rakkolainen, Ahmed Farooq, Jari Kangas, Jaakko Hakulinen, Jussi Rantala, Markku Turunen, and Roope Raisamo. 2021. Technologies for multimodal interaction in extended reality—a scoping review. *Multimodal Technologies and Interaction*, 5(12):81.
- Santhosh K. Ramakrishnan and et al. 2021. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv:2109.08238*.
- Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2024a. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.
- Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nico-

- Ias Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2024b. Sam 2: Segment anything in images and videos.
- MARNIE RENDA and JENNIFER E. LAPE. 2018. Feasibility and Effectiveness of Telehealth Occupational Therapy Home Modification Interventions. *International Journal of Telerehabilitation*, 10(1):3–14.
- Rev.com. Transcribe Speech to Text.
- Flavio Ribeiro, Dinei Florencio, Philip A Chou, and Zhengyou Zhang. 2012a. Auditory augmented reality: Object sonification for the visually impaired. In *2012 IEEE 14th international workshop on multimedia signal processing (MMSP)*, pages 319–324. IEEE.
- Flávio Ribeiro, Dinei Florêncio, Philip A. Chou, and Zhengyou Zhang. 2012b. Auditory augmented reality: Object sonification for the visually impaired. In *2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)*, pages 319–324.
- Anthony E. Richardson, Daniel R. Montello, and Mary Hegarty. 1999. Spatial knowledge acquisition from maps and from navigation in real and virtual environments. *Memory and Cognition*, 27(4):741–750.
- Marvelmind Robotics. 2024. Automated scanning and inspection. Accessed: 2024-08-15.
- Agnieszka Roginska and Paul Geluso. 2017. *Immersive Sound*. Focal Press.
- Sergio Romero, Mi Lee, Ivana Simic, Charles Levy, and Jon Sanford. 2017. Development and validation of a remote home safety protocol. *Disability and Rehabilitation: Assistive Technology*, 13:1–7.
- Dariusz Rumiński. 2015. An experimental study of spatial sound usefulness in searching and navigating through ar environments. *Virtual Reality*, 19(3-4):223–233.
- Manaswi Saha, Siddhant Patil, Emily Cho, Evie Yu-Yen Cheng, Chris Horng, Devanshi Chauhan, Rachel Kangas, Richard McGovern, Anthony Li, Jeffrey Heer, and Jon E. Froehlich. 2022. Visualizing Urban Accessibility: Investigating Multi-Stakeholder Perspectives through a Map-based Design Probe Study. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI ’22, pages 1–14, New York, NY, USA. Association for Computing Machinery.

- Manaswi Saha, Michael Saugstad, Hanuma Teja Maddali, Aileen Zeng, Ryan Holland, Steven Bower, Aditya Dash, Sage Chen, Anthony Li, Kotaro Hara, and Jon Froehlich. 2019. Project Sidewalk: A Web-based Crowdsourcing Tool for Collecting Sidewalk Accessibility Data At Scale. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, Glasgow Scotland Uk. ACM.
- Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. 2019. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9339–9347.
- ByteDance Seed. 2025. Seed 1.6. https://seed.bytedance.com/en/seed1_6/. Accessed: 2025-11-12.
- Andrés Serna and Beatriz Marcotegui. 2013. Urban accessibility diagnosis from mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 84:23–32.
- Dhruv Shah and et al. 2023. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *CoRL (PMLR)*.
- Roy Sheffer and Yossi Adi. 2023. I hear your true colors: Image guided audio generation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugging-gpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*.
- Xiangxi Shi, Zhonghua Wu, and Stefan Lee. 2024. Aware visual grounding in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14056–14065.
- Xinyu Shi, Yinghou Wang, Ryan Rossi, and Jian Zhao. 2025. Brickify: Enabling expressive design intent specification through direct manipulation on design tokens. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

- Gaurav Shinde, Aniruddha Ravi, Esha Dey, S. Sakib, Manali Rampure, and Nilanjan Roy. 2025. A survey on efficient vision-language models. *WIREs Data Mining and Knowledge Discovery*, 15(3).
- Ben Shneiderman. 2020. Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human-Computer Interaction*, 36(6):495–504.
- Alexander W. Siegel and Sheldon H. White. 1975. The development of spatial representations of large-scale environments. In *Advances in Child Development and Behavior*, volume 10, pages 9–55. Academic Press.
- Walter C. S. S. Simões, Guido S. Machado, André M. A. Sales, Mateus M. de Lucena, Nasser Jazdi, and Vicente F. de Lucena. 2020. A review of technologies and techniques for indoor navigation systems for the visually impaired. *Sensors*, 20(14).
- Mustafa Sirakaya and Didem Alsancak Sirakaya. 2022. Augmented reality in stem education: A systematic review. *Interactive Learning Environments*, 30(8):1556–1569.
- Yadurshana Sivashankar, Philip He, Patrick Tsapoitis, Evan Skorski, and Myra A. Fernandes. 2025. The necessity of motoric engagement in enhancing route memory. *British Journal of Psychology*, 116(4):1164–1188.
- Skydio. 2025. Introducing 3d indoor capture. Accessed: 2025-02-14.
- Stanley K. Smith, Stefan Rayer, and Eleanor A. Smith. 2008. Aging and Disability: Implications for the Housing Industry and Housing Policy in the United States. *Journal of the American Planning Association*, 74(3):289–306. Publisher: Routledge _eprint: <https://doi.org/10.1080/01944360802197132>.
- Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. 2025. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12078–12088.
- Edward Steinfeld, Danise R. Levine, and Scott M. Shea. 1998. Home modifications and the fair housing law. *Technology and Disability*, 8(1-2):15–35. Publisher: IOS Press.
- Ilias Stogiannidis, Steven McDonagh, and Sotirios A. Tsafaris. 2025. Mind the gap: Benchmarking spatial reasoning in vision-language models. *arXiv preprint arXiv:2503.19707*.

- Julian Striegl, Claudia Loitsch, Jan Schmalfuss-Schwarz, and Gerhard Weber. 2020. Analysis of indoor maps accounting the needs of people with impairments. In *Computers Helping People with Special Needs*, pages 305–314, Cham. Springer International Publishing.
- Linda Struckmeyer, Jane Morgan-Daniel, Sherry Ahrentzen, and Carlyn Ellison. 2021. Home Modification Assessments for Accessibility and Aesthetics: A Rapid Review. *HERD: Health Environments Research & Design Journal*, 14(2):313–327.
- Peter Sturm. 2021. Pinhole camera model. In *Computer Vision: A Reference Guide*, pages 983–986. Springer.
- Xia Su, Daniel Campos Zamora, and Jon E Froehlich. 2024a. Rais: Towards a robotic mapping and assessment tool for indoor accessibility using commodity hardware. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–5.
- Xia Su, Ruiqi Chen, Benlin Liu, Jingwei Ma, Zonglin Di, Ranjay Krishna, and Jon Froehlich. 2026a. Capnav: Benchmarking vision language models on capability-conditioned indoor navigation. *arXiv preprint arXiv:2602.18424*.
- Xia Su, Ruiqi Chen, Jingwei Ma, Chu Li, and Jon E Froehlich. 2025a. Flymethrough: Human-ai collaborative 3d indoor mapping with commodity drones. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14.
- Xia Su, Ruiqi Chen, Richard Weiye Zhang, Jingwei Ma, and Jon E. Froehlich. 2024b. A demo of diam: Drone-based indoor accessibility mapping. In *Extended Abstract Proceedings of the 2024 ACM Symposium on User Interface Software and Technology*.
- Xia Su, Kaiming Cheng, Han Zhang, Jaewook Lee, and Jon E. Froehlich. 2022. Towards Semi-automatic Detection and Localization of Indoor Accessibility Issues using Mobile Depth Scanning and Computer Vision. ArXiv:2210.02533 [cs].
- Xia Su, Kaiming Cheng, Han Zhang, Jaewook Lee, Wyatt Olson, and Jon E. Froehlich. 2023. A Demonstration of RASSAR: Room Accessibility and Safety Scanning in Augmented Reality. In *The 25th Inter-*

- national ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–4, New York NY USA. ACM.
- Xia Su, Jon E Froehlich, Eunye Koh, and Chang Xiao. 2024c. Sonifyar: Context-aware sound generation in augmented reality. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13.
- Xia Su, Cuong Nguyen, Matheus A Gadelha, and Jon E Froehlich. 2026b. Depthscape: Authoring 2.5 d designs via depth estimation, semantic understanding, and geometry extraction. In *Proceedings of the 2026 Designing Interactive Systems Conference*, pages 1–16.
- Xia Su, Cuong Nguyen, Matheus A Gadelha, Yu Shen, Stefano Petrangeli, and Jon E. Froehlich. 2025b. Authoring 2.5d designs with depth estimation. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '25, New York, NY, USA. Association for Computing Machinery.
- Xia Su, Han Zhang, Kaiming Cheng, Jaewook Lee, Qiaochu Liu, Wyatt Olson, and Jon E Froehlich. 2024d. Rassar: Room accessibility and safety scanning in augmented reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–17.
- Ryo Suzuki, Rubaiat Habib Kazi, Li-yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. Realitysketch: Embedding responsive graphics and visualizations in ar through dynamic sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 166–181, New York, NY, USA. Association for Computing Machinery.
- Ellen D. Taira and Jodi Carlson. 2014. *Aging in Place: Designing, Adapting, and Enhancing the Home Environment*. Routledge, New York.
- Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. 2023. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–12.
- Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng

Yan. 2024. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. *arXiv preprint arXiv:2412.06974*.

Holly A. Taylor and Barbara Tversky. 1992. Spatial mental models derived from survey and route descriptions. *Journal of Memory and Language*, 31(2):261–292.

Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, Bowen Shen, Zihan Zhang, Zihan Jiang, Zhixian Zheng, Zhichao Song, Zhenbo Luo, Yue Yu, Yudong Wang, Yuanyuan Tian, Yu Tu, Yihan Yan, Yi Huang, Xu Wang, Xinzhe Xu, Xingchen Song, Xing Zhang, Xing Yong, Xin Zhang, Xiangwei Deng, Wenyu Yang, Wenhan Ma, Weiwei Lv, Weiwei Zhuang, Wei Liu, Sirui Deng, Shuo Liu, Shimao Chen, Shihua Yu, Shaohui Liu, Shande Wang, Rui Ma, Qiantong Wang, Peng Wang, Nuo Chen, Menghang Zhu, Kangyang Zhou, Kang Zhou, Kai Fang, Jun Shi, Jinhao Dong, Jiebao Xiao, Jiaming Xu, Huaqiu Liu, Hongshen Xu, Heng Qu, Haochen Zhao, Hanglong Lv, Guoan Wang, Duo Zhang, Dong Zhang, Di Zhang, Chong Ma, Chang Liu, Can Cai, and Bingquan Xia. 2025a. Mimo-vl technical report.

Qwen Team. 2025. Qwen3 technical report.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihang Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Jiazheng Xu, Jiale Zhu, Jiali Chen, Jing Chen, Jinhao Chen, Jinghao Lin, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Sheng Yang, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Shangqin Tu, Shengbiao Meng, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wenkai Li, Wei Jia, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyue Fan, Xuancheng Huang, Yanling Wang, Yadong Xue, Yanfeng Wang, Yanzi Wang, Yifan An, Yifan Du, Yiming Shi, Yiheng Huang, Yilin Niu, Yuan Wang, Yuanchang Yue, Yuchen Li, Yutao Zhang, Yuting Wang, Yu Wang, Yuxuan Zhang, Zhao Xue, Zhenyu Hou, Zhengxiao Du, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang,

- Yuxiao Dong, and Jie Tang. 2025b. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning.
- Rebuilding Together. Safe at Home Checklist.
- Edward C. Tolman. 1948. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208.
- Machiko R. Tomita, Sumandeep Saharan, Sheela Rajendran, Susan M. Nochajski, and Jo A. Schweitzer. 2014. Psychometrics of the Home Safety Self-Assessment Tool (HSSAT) to Prevent Falls in Community-Dwelling Older Adults. *The American Journal of Occupational Therapy*, 68(6):711–718.
- P. van Turenout, G. Honderd, and L. van Schelven. 1992. Wall-following control of a mobile robot. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 280,281,282,283,284,285, Los Alamitos, CA, USA. IEEE Computer Society.
- Barbara Tversky. 1993. Cognitive maps, cognitive collages, and spatial mental models. In *Spatial Information Theory: A Theoretical Basis for GIS*, volume 716 of *Lecture Notes in Computer Science*, pages 14–24. Springer.
- Barbara Tversky and Paul U. Lee. 1999. Pictorial and verbal tools for conveying routes. In *Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science*, volume 1661 of *Lecture Notes in Computer Science*, pages 51–64. Springer.
- Jakob von Uexküll. 2010. *A Foray into the Worlds of Animals and Humans: With a Theory of Meaning*. University of Minnesota Press, Minneapolis, MN.
- Ultralytics. 2023. YOLOv8: Cutting-edge object detection. <https://github.com/ultralytics/ultralytics>. Accessed: 2025-04-06.
- Ultralytics. Accessed September 11, 2023. YOLOv5 github repository. <https://github.com/ultralytics/yolov5>.
- Unity Technologies. 2023. Getting started with unity mars. <https://unity.com/products/mars/get-started>. Accessed: 2024-04-02.

- Paul Upchurch and Ransen Niu. 2022. A Dense Material Segmentation Dataset for Indoor and Outdoor Scene Parsing. ArXiv:2207.10614 [cs].
- Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. 2025a. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5294–5306.
- Liuyi Wang, Xinyuan Xia, Hui Zhao, Hanqing Wang, Tai Wang, Yilun Chen, Chengju Liu, Qijun Chen, and Jiangmiao Pang. 2025b. Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9455–9465.
- Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. 2024a. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision.
- Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. 2024b. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709.
- Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. 2024c. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*.
- Ethan Weber, Norman Müller, Yash Kant, Vasu Agrawal, Michael Zollhöfer, Angjoo Kanazawa, and Christian Richardt. 2025. Fillerbuster: Multi-view scene completion for casual captures. ArXiv:2502.05175.
- Galen Weld, Esther Jang, Anthony Li, Aileen Zeng, Kurtis Heimerl, and Jon E. Froehlich. 2019. Deep Learning for Automatically Detecting Sidewalk Accessibility Problems Using Streetscape Imagery. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pages 196–209, Pittsburgh PA USA. ACM.
- Maximilian Windl, Patryk Z. Laboda, and Sven Mayer. 2025. Designing effective consent mechanisms for

- spontaneous interactions in augmented reality. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–18.
- Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon E. Froehlich. 2011. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing*, 3(3):1–27.
- Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.
- Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. 2025. Spatial-mlm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*.
- Mingrui Wu, Zhaozhi Wang, Fangjinhua Wang, Jiaolong Yang, Marc Pollefeys, and Tong Zhang. 2026. From indoor to open world: Revealing the spatial reasoning gap in mllms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16789–16799.
- Ziyang Xie, Zhizheng Liu, Zhenghao Peng, Wayne Wu, and Bolei Zhou. 2025. Vid2sim: Realistic and interactive simulation from video for urban navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1581–1591.
- Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. 2024a. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*.
- Peiran Xu, Sudong Wang, Yao Zhu, Jianing Li, and Yunjian Zhang. 2025. Spatialbench: Benchmarking multimodal large language models for spatial cognition. *arXiv preprint arXiv:2511.21471*.
- Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. 2024b. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*.
- Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. 2025a. Kwai keye-vl 1.5 technical report. *arXiv preprint arXiv:2509.01563*.

- Fuxiang Yang, Donglin Di, Lulu Tang, Xuancheng Zhang, Lei Fan, Hao Li, Chen Wei, Tonghua Su, and Baorui Ma. 2026. Chain of world: World model thinking in latent motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6675–6684.
- Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2025b. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10632–10643.
- Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024a. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381.
- Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024b. Depth anything v2. *arXiv preprint arXiv:2406.09414*.
- Xingbin Yang, Liyang Zhou, Hanqing Jiang, Zhongliang Tang, Yuanbo Wang, Hujun Bao, and Guofeng Zhang. 2020. Mobile3drecon: Real-time monocular 3d reconstruction on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3446–3456.
- Chris Yoon, Ryan Louie, Jeremy Ryan, MinhKhang Vu, Hyegi Bang, William Derksen, and Paul Ruvolo. 2019. Leveraging augmented reality to create apps for people with visual disabilities: A case study in indoor navigation. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '19*, page 210–221, New York, NY, USA. Association for Computing Machinery.
- Han Yue and Hangbin Wu. 2018. Update method of indoor maps based on volunteered geographic information (vgi). In *2018 26th International Conference on Geoinformatics*, pages 1–5. IEEE.
- Hao Xuan Zhang, Yilin Yang, and Zhengbo Zou. 2024. Icon drone: Autonomous indoor exploration using unmanned aerial vehicle for semantic 3d reconstruction. In *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 66–76.
- Ruowen Zhao, Zhengyi Wang, Yikai Wang, Zihan Zhou, and Jun Zhu. 2024. Flexidreamer: Single image-to-3d generation with flexicubes. *arXiv preprint arXiv:2404.00987*.

- Yuhang Zhao, Elizabeth Kupferstein, Brenda Veronica Castro, Steven Feiner, and Shiri Azenkot. 2019. 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*, pages 387–402, New Orleans LA USA. ACM.
- Gengze Zhou, Yicong Hong, and Qi Wu. 2023. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv:2305.16986*.
- Guyue Zhou, Ang Liu, Kang Yang, Tao Wang, and Zexiang Li. 2014. An embedded solution to visual mapping for consumer drones. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 656–661.
- Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L. Berg. 2018. Visual to sound: Generating natural sound for videos in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- ZhiYing Zhou, Adrian David Cheok, Yan Qiu, and Xubo Yang. 2007. The Role of 3-D Sound in Human Reaction and Performance in Augmented Reality Environments. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 37(2):262–272. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans.
- Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. 2025. Internv13: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.
- Ruijie Zhu, Ziyang Song, Li Liu, Jianfeng He, Tianzhu Zhang, and Yongdong Zhang. 2024. Ha-bins: Hierarchical adaptive bins for robust monocular depth estimation across multiple datasets. *IEEE Transactions on Circuits and Systems for Video Technology*, 34:4354–4366.
- Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee,

Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. 2023. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR.