PathSense: Affordable Smartphone-Based Assistive Navigation Using LiDAR and Haptic Feedback

An Engineering Design Research Paper

Noah Hellyer and Aryaveer Sheoran

Cherry Creek High School
9300 E. Union Ave., Greenwood Village, CO 80111

Adult Sponsor: Ethan Dusto

Regeneron International Science and Engineering Fair
Category: Systems Software / Biomedical Engineering

Abstract

PathSense is an experimental, chest-mounted smartphone navigation aid designed to supplement established mobility practices for blind and low-vision travelers. The active system combines LiDAR depth, ARKit visual–inertial odometry, an on-device YOLO11n detector, local terrain classification, route context, and distance-encoded audio pulses that prototype a planned haptic-shoe interface. Failure-driven engineering exposed recurring hazards: missing depth was confused with a drop, a global floor model failed on sloped pavement, moving people were treated as static terrain, coordinate-sign errors inverted steering, and retained camera frames stalled the sensing pipeline. These failures produced conservative rules and repeatable regression cases. The research stack preserves vertical occupancy in GuideSpace, sweeps a configured body volume through candidate motions, predicts dynamic conflicts, latches curb boundaries, and maintains multiple route-region hypotheses. Further default-off components model uncertain cue response, require maneuver witnesses, recommend decision-relevant scans, budget attention, and store consented local route memory. Camera semantics may annotate occupied evidence but may never clear LiDAR occupancy; route preference may rank only candidates already admitted by geometric, topological, and crossing constraints. Every research component publishes a typed result or refusal into synchronized replay and remains shadow-only: it cannot steer, speak, or grant crossing permission. The current verification run passed 1,518 simulator tests, skipped one environment-gated integration test, and failed none. Compiler warnings remain, and no participant or sustained-device evidence is claimed. The paper therefore reports architecture, algorithms, failure analysis, and a staged evaluation framework rather than clinical efficacy.

Contents

1 Introduction **4**

2 Background and Related Work **5**

2.1 Existing mobility aids and their gaps 5

2.2 Evidence that real-time feedback works 5

2.3 The smartphone LiDAR opportunity 6

2.4 Uncertainty-aware navigation 6

3 Research Question, Hypothesis, and Engineering Goals **6**

3.1 Research question 6

3.2 Hypothesis 7

3.3 Engineering design criteria 7

4 Materials **8**

5 Methods: Iterative Engineering Design **8**

5.1 Development methodology 8

5.2 Generation 1: the naïve depth heuristic and why it failed 9

5.3 Generation 2: why a global calibrated plane failed 9

5.4 Generation 3: the world-anchored terrain grid 11

5.5 Small steps: when absolute height physically cannot work 12

5.6 Feedback design: pulse audio as a haptic-shoe prototype 12

5.7 Performance: making it survive an iPhone 14 Pro 14

5.8 Truthful drop-offs: the war on the false urgent alert 14

5.9 Stairs: detecting a structure, not a surface 15

5.10 People are not terrain 17

5.11 The mirrored-steering defect: why conventions need tests 18

5.12 Navigation without a compass 18

5.13 Intersection safety as an evidence-gated state machine 19

5.14 The semantic object layer: labels annotate, never veto 20

5.15 GuideSpace: vertical evidence and body-relative risk 22
5.16 Coordinated research stack: route intent to one action 24
5.17 Local path planning and the planner view 30

6 System Architecture and Experimental Layers **30**

7 Failure Catalog **31**

8 Verification and Research Instrumentation **36**
8.1 Automated verification 36
8.2 Field replay: turning walks into experiments 37
8.3 Trial instrumentation 38

9 Results **39**

10 Planned Human-Participant Evaluation **40**

11 Discussion **42**
11.1 What the failures teach 42
11.2 Relation to prior work 42
11.3 Limitations 43
11.4 Future work 43

12 Conclusion **44**

1 Introduction

Globally, at least 2.2 billion people have a vision impairment, and in at least one billion of those cases the impairment could have been prevented or has yet to be addressed [1]. The Vision Loss Expert Group estimated that in 2015 approximately 253 million people were visually impaired—36 million of them blind—and that 89% of visually impaired people live in low- and middle-income countries [2]. For this population, independent travel is one of the most consequential and most dangerous activities of daily life. Navigation is hardest exactly where it matters most: in unfamiliar environments, where a traveler cannot rely on memorized routes [3]. Surveys of blind and low-vision pedestrians report that a majority have experienced mobility-related accidents; El-Taher et al. report that 63% of surveyed travelers with visual impairments had been injured while navigating outdoors [4], consistent with earlier accident surveys in which head-level and trip hazards dominated [5].

The long cane and the guide dog remain the primary mobility aids, and both are effective at short-range obstacle contact—but neither provides navigation ("turn left in fifty feet"), neither detects hazards beyond arm's or harness's reach, and neither can name what lies ahead. Electronic travel aids (ETAs) have been studied for six decades, yet reviews repeatedly reach the same conclusions: most systems never leave the laboratory, battery life and wearability are poor, extensive training is required, and cost puts them out of reach of the very population that needs them most [3, 6]. Encouragingly, a randomized clinical trial showed that a wearable collision-warning device with vibrotactile wristbands reduced collisions by 37% relative to the cane or guide dog alone [7]—evidence that real-time sensing genuinely improves safety when it is delivered in a usable form. That device, however, was dedicated research hardware providing warning only, without terrain understanding or wayfinding.

Two technical developments make a different approach possible. First, Apple's Pro-line iPhones combine a solid-state LiDAR scanner, a camera array, and a dedicated neural processor (the Apple Neural Engine) [8, 9]—sensor hardware comparable to early research ETAs, already in the pockets of millions of users, including blind users, for whom the iPhone is the dominant accessible smartphone platform. Second, modern lightweight object detectors such as YOLO11n [10] can run continuously on that neural processor without occupying the CPU and GPU needed for real-time geometry processing. Prior surveys explicitly identify the smartphone as the most promising ETA platform because of its processing power, ubiquity, and cost trajectory [3], and haptics research shows that vibration-coded feedback is language-independent, private, and usable in noisy streetscapes where speech and sonification fail [11].

This paper reports the design, iterative engineering, failure analysis, and verification of PathSense, a smartphone application that turns a chest-mounted iPhone into a combined terrain sensor, obstacle

classifier, and navigation guide, with feedback delivered as distance-encoded audio pulse patterns explicitly designed to map one-to-one onto a planned haptic-shoe interface. Unlike a conventional report of a finished artifact, we document *why each earlier design failed*: the research archive preserves abandoned or superseded architectures and specific field failures, each with a diagnosed root cause and, where possible, a regression test. We argue that this failure catalog, the conservative evidence rules it produced, and the separation of active and experimental safety layers are the project's most transferable engineering contributions.

2 Background and Related Work

2.1 Existing mobility aids and their gaps

Real and Araujo's survey of navigation systems for the blind traces ETAs from early sonar torches to modern SLAM-based prototypes and identifies three required functional blocks—positioning, environmental monitoring, and the user interface—while noting that cognitive overload, not sensing, is the usual failure mode: a system cannot stream raw spatial complexity through hearing or touch without overwhelming the traveler [3]. Messaoudi et al. add that many published systems are "lab-based," optimized for accuracy over wearability, demand long training, or have limited battery life [6]. El-Taher et al. survey outdoor navigation applications specifically and find that existing apps lack real-time obstacle detection and environmental detail [4]. Commercially available smart canes, ultrasonic wearables, and camera glasses cost from hundreds to several thousand dollars, and a trained guide dog costs tens of thousands of dollars to raise and train—costs largely borne by charities with long waiting lists. The affordability gap is structural: any solution requiring dedicated hardware inherits a dedicated-hardware price.

2.2 Evidence that real-time feedback works

The strongest causal evidence comes from Pundlik et al.'s randomized clinical trial of a chest-mounted camera device with vibrating wristbands: in home use by 31 blind and visually impaired participants, contacts with obstacles fell 37% relative to their conventional aid alone [7]. Khusro et al. demonstrated that blind users can reliably discriminate vibration patterns (65–90% recognition across ten patterns) and argue that haptic coding is preferable to speech in noisy environments and to sonification for privacy and audibility [11]. Together these results motivated two PathSense decisions: feedback should be continuous and pattern-coded rather than spoken wherever possible, and speech should be reserved for information that patterns cannot carry (nouns, route instructions,

urgent stops).

2.3 The smartphone LiDAR opportunity

Apple's ARKit exposes a 256×192 depth map at up to 60 Hz from the LiDAR scanner, fused camera–IMU visual–inertial odometry (world tracking) with centimeter-scale relative accuracy, plane detection, and gravity-aligned camera transforms [8]. The Apple Neural Engine executes quantized convolutional networks such as YOLO11n [10], trained on the COCO dataset [12], at interactive rates on silicon physically separate from the CPU/GPU [9]. Contemporary systems such as NaviSense combine LiDAR, augmented reality, vision-language perception, and audio–haptic feedback, demonstrating that multimodal smartphone assistance is now an active research area [13]. PathSense is differentiated not by claiming the first such combination, but by its terrain-specific failure analysis, asymmetric evidence authority, body-height risk representation, and replay-driven safety workflow.

2.4 Uncertainty-aware navigation

Learning-based perception can increase semantic range, but a probability attached to a label is not a guarantee of path clearance. Recent work applies statistical calibration and conformal prediction to navigation decisions [14, 15]. This motivates PathSense to represent unknown space explicitly and to keep learned labels subordinate to metric occupancy. The default-off Action–Path Envelope (Section 5.15) already applies deterministic body-swept clearance, freshness, dynamic, and topological gates in shadow. Its conformal extension remains uncalibrated and neither version is an active steering component.

3 Research Question, Hypothesis, and Engineering Goals

3.1 Research question

Within a defined set of indoor and outdoor pedestrian environments, does adding PathSense to a blind or low-vision traveler's usual mobility aid reduce environmental contacts and navigation errors without increasing course time, unsafe commands, or cognitive workload?

3.2 Hypothesis

Building on evidence that vibrotactile collision warnings reduce obstacle contacts [7] and that pattern-coded haptic feedback is discriminable and non-intrusive [11], we hypothesize that a smartphone system combining (a) metric terrain geometry from LiDAR, (b) semantic object awareness from an on-device detector, (c) explicit unknown-space handling, and (d) distance-encoded pulse feedback will reduce environmental contacts and navigation errors when used *with* a traveler's usual mobility aid. The primary outcomes are contacts and near-misses per unit distance, wrong turns, intervention rate, and successful completion; course time and cognitive workload are guardrails. The present paper reports the engineering phase that makes this hypothesis testable; controlled human-participant trials are designed (Section 10) but not yet conducted.

3.3 Engineering design criteria

The hypothesis imposes quantified design criteria; each is verified in Section 9.

G1. Real time on consumer hardware. Maintain walking-speed responsiveness on an iPhone 14 Pro, monitor processing age and dropped frames, and degrade conservatively under thermal pressure.

G2. Trustworthy alerts. Urgent alerts (*stop*, *drop ahead*) must be near-zero false positive, because false alarms destroy user trust. Missing observations remain explicitly unknown; unresolved risk must produce slow, stop, or scan behavior rather than an unsupported "clear" or speculative steering cue.

G3. Terrain completeness. Detect obstacles (≥30 cm), curbs and steps (10–30 cm), small steps (4–15 cm), drop-offs (≥15 cm), ascending and descending staircases, and moving pedestrians—and distinguish look-alikes: a curb from a trash can, a staircase from a wall, a pothole from a drop-off.

G4. Navigation, not just warning. Provide GPS walking routes, body-relative turn cues without a usable magnetometer, an audio homing beacon, and on-demand environmental descriptions.

G5. Low incremental cost and privacy-preserving operation. No dedicated sensing computer beyond a supported phone, chest mount, and audio or haptic output; all live inference on-device; no network required for guidance. Research uploads are optional and consent-gated.

G6. Measurability. Built-in instrumentation sufficient to quantify "safer" (collisions, near-misses, wrong turns, time, walking speed) while preserving independent observer adjudication.

4 Materials

- **Target device:** Apple iPhone 14 Pro (A16 Bionic; LiDAR scanner; 256×192 scene-depth stream with per-pixel confidence; 1920×1440 RGB capture), carried in a portrait chest mount. Earbuds for stereo audio output.
- **Software platform:** Swift / SwiftUI in Xcode; frameworks: ARKit (world tracking, scene depth, plane anchors), RealityKit (GPU camera rendering and planner visualization), Vision + Core ML (object detection), AVFoundation (speech synthesis and generated pulse audio), MapKit + CoreLocation (walking routes, GPS), and an optional provider-neutral Supabase transport for consented research reports.
- **Perception model:** YOLO11n (≈2.6 M parameters) [10] pretrained on COCO [12], exported to Core ML (square 640 input, FP16, embedded non-maximum suppression) and pinned to the Apple Neural Engine; a larger YOLO11x "teacher" model runs offline for auto-labeling collected field data (toolchain pinned to `torch` 2.5.1, `coremltools` 8.x, Python 3.11).
- **Verification tooling:** 134 Swift test files containing 1,519 `@Test` declarations; a legacy depth replay format; and a synchronized ClipV2 format for depth, confidence, pose, motion, route, guidance, events, compressed RGB, rendered-cue response episodes, and typed perceptual-navigation results. A shadow harness compares baseline and candidate decisions over the same recorded input, while subsystem-scoped evidence gates reject incomplete or incompatible streams.
- **Experimental controls:** Per-session feature freezing, system-health monitoring, a traveler body profile, and explicit evidence-source tags support reproducible comparisons between active and shadow systems.

5 Methods: Iterative Engineering Design

5.1 Development methodology

PathSense was developed by an explicit engineering-design loop: *specify* → *plan* → *implement* → *unit-test* → *field-walk* → *capture failures* → *diagnose* → *fix* → *pin with a regression test*. Three practices distinguish the process:

1. **Design documents before code.** Major subsystems (stair detection, planner object segmentation, the semantic object layer) each began as a written specification enumerating the problem,

candidate approaches, and *explicitly rejected alternatives* with reasons—so later sections of this paper can quote why an approach was not taken, not merely that it was not.

2. **Field walks as experiments.** Every capability was walked outdoors on real sidewalks, slopes, stairs, and streets. Failures observed on a walk were recorded (eventually as replayable depth clips), diagnosed to a root cause, fixed, and converted into automated tests, so no failure could silently return.

3. **Replay-driven evaluation.** Once the clip recorder existed, judging a tuning change no longer required re-walking a course: the shadow-replay harness re-runs recorded walks under baseline and candidate configurations and reports every decision divergence (Section 8).

The remainder of this section narrates the design generations in order, because the failures of each generation are the rationale for the next. Table 1 summarizes; Table 4 catalogs individual failures with root causes and fixes.

5.2 Generation 1: the naïve depth heuristic and why it failed

The first prototype colorized the LiDAR depth map over the live camera feed and divided the frame into left, center, and right regions, speaking "stop," "move left," or "move right" from the ratio of near pixels in each region. It failed within the first indoor walks: *the floor is the largest near surface in every frame*, so ordinary ground triggered constant stops. A first patch used the gravity-aligned camera transform to discard pixels more than 0.3 m below the camera—but this exposed the deeper problem: a region-ratio detector has no concept of terrain. It cannot distinguish a wall from a curb from a descending staircase from a pothole, cannot remember a hazard once it leaves the field of view, and cannot say how far away anything is with respect to the walking surface. Assistive guidance needs *terrain understanding*, not pixel statistics.

5.3 Generation 2: why a global calibrated plane failed

The second design added a chest-mount calibration step and tracked a single RANSAC-fitted [16] ground plane, classifying hazards as deviations from that plane, with a seven-signal confidence tracker driving recalibration. It was replaced for two reasons that shaped everything after. First, per-user calibration is a usability defect: mount height varies across users and even across a single walk, so any assumed height eventually lies (the final system instead estimates camera height continuously with an exponential moving average, accepting 0.5–2.2 m). Second, one global plane is the wrong model of a street: real pavement is crowned, sloped, and stepped, so a single plane

Table 1: Architecture generations and their research status.

Generation	Approach	Outcome and current status
1. Region-scored depth heuristic	Ratio of near pixels in left/center/right thirds of the raw depth map; spoken stop/left/right	Failed: the floor itself read as an obstacle; no terrain semantics (curbs, steps, drops); superseded
2. Calibrated ground-plane detector	Per-user chest-mount calibration; RANSAC ground plane tracked as single reference; hazards = deviations from the plane	Abandoned: per-user calibration brittle; one global plane cannot represent crowned, sloped, or stepped streets
3. World-anchored terrain grid	Depth back-projected into a persistent top-down 7.5 cm grid fused from plane anchors, mesh, and depth-RANSAC; multi-pass per-cell classifier; hazards remembered in world space	Active. Every later capability (small steps, stairs, people, drop-off truthfulness, planning) is a refinement of this grid
3+. Semantic object layer	YOLO11n on the Apple Neural Engine fused with LiDAR geometry; labels annotate the grid's hazards	Adopted with the safety rule *camera semantics never clear LiDAR occupancy*; extends awareness and names obstacles
4. GuideSpace research layer	World-anchored 10 cm vertical evidence bins, high-confidence free-space rays, semantic evidence, and traveler-relative body slices	Implemented in shadow mode and consumed by body-volume candidate evaluation; does not control active guidance
5. Integrated perceptual-navigation research layer	Curb-latched topological boundaries, accessible route graph, multi-hypothesis route belief, maneuver witnesses, decision-relevant scans, guided-response envelopes, attention policy, and consented route memory	Implemented as a default-off, typed shadow stack with replay and adversarial tests. Some live producers and calibration artifacts remain unavailable; the stack cannot steer, speak, or authorize crossings

leaves residuals that grow with distance and read as phantom bumps. The lesson—*local* ground models beat global ones—returned twice more (Sections 5.5 and 5.8).

5.4 Generation 3: the world-anchored terrain grid

The surviving architecture back-projects each depth pixel through the camera intrinsics and the ARKit world transform into 3-D world coordinates. A depth pixel (u, v) with measured depth $d(u, v)$ lifts to a camera-space point and then to world space by

$$\mathbf{p}_c \;=\; d(u,v)\, K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \qquad \tilde{\mathbf{p}}_w \;=\; T_{wc}\, \tilde{\mathbf{p}}_c, \tag{1}$$

where K is the camera intrinsic matrix rescaled to the 256×192 depth resolution and $T_{wc} \in SE(3)$ is ARKit's gravity-aligned camera-to-world transform. Points are binned into a persistent top-down grid of 7.5 cm cells keyed by *world* position (so cell history survives as the user moves), covering -0.4 to 4.0 m ahead and ± 1.0 m laterally (59×27 cells).

The walking surface is a fused *FloorPlane* $(\mathbf{n}, d_0)$—combining ARKit floor-classified plane anchors with RANSAC [16] over trusted depth samples—and every cell's height is measured against it, $h(\mathbf{p}_w) = \mathbf{n} \cdot \mathbf{p}_w + d_0$. The RANSAC stage draws three-point minimal samples with a 2.5 cm inlier band for 60 iterations; since a sample succeeds only if all three points are floor inliers, the probability of finding the plane is $1 - (1 - w^3)^{60}$ for inlier ratio w, i.e. >99.9% at $w = 0.5$ and 98.1% even at $w = 0.4$—which is why 60 iterations, chosen for CPU budget, are statistically sufficient.

Each cell keeps a 16-deep ring buffer of height samples summarized by *robust* statistics—the median $\tilde{h}$ and the median absolute deviation,

$$\mathrm{MAD} \;=\; \mathrm{median}_i \big|\, h_i - \tilde{h}\,\big|, \qquad \hat{\sigma} \approx 1.4826\,\mathrm{MAD}, \tag{2}$$

rather than mean and standard deviation: with a 50% breakdown point, the median tolerates up to half of a cell's samples being flying-pixel or multipath outliers before its estimate moves, whereas a single outlier corrupts a mean. A multi-pass classifier labels each cell (floor, bump, small step up/down, step up, obstacle, drop-off, stairs, person, raised floor), clusters same-class cells into hazards, and a world-space hazard map remembers confirmed hazards for 25 s—so a trash can passed at arm's length is still avoided when it leaves the camera's view. All guidance derives from a single decision function over this grid, with one command spoken or pulsed at a time under a strict priority order (drop-off 100, stop 95, turns 90, stairs 74–76, obstacle 70, . . . , clear 0).

Generation 3 was not a finished system but a platform. The following subsections describe the major capabilities added on top of it, each driven by a documented failure of the system as it existed.

5.5 Small steps: when absolute height physically cannot work

Sidewalk lips, door thresholds, and broken slabs of 4–12 cm are among the most common trip hazards [5], but they are *undetectable as absolute heights* against a fitted floor plane: outdoor cross-slope alone produces ±8 cm of legitimate height variation at 4 m range, swamping the signal. The fix reframed the problem locally: a small step is a *height discontinuity between adjacent grid rows*, detected only within a 2 m near band where LiDAR noise permits, with a noise-adaptive threshold

$$\tau_{\text{edge}} \;=\; \max\!\big(0.04\ \text{m},\ 4.0 \times \widetilde{\sigma}_{\text{band}}\big), \qquad \widetilde{\sigma}_{\text{band}} = \text{median}\{\text{MAD of floor cells in the 4-row band}\}, \tag{3}$$

requiring flat plateaus on both sides of the edge and coherence across at least three adjacent columns (a real step edge crosses the walking corridor; a noise spike does not). Where a band contains too few floor cells to measure its noise empirically, the threshold falls back to an analytic model that propagates the LiDAR's range noise, $\sigma(d) = 0.008 + 0.012\,d$ meters (fit empirically to static-scene captures), onto cell heights through the grazing-ray geometry:

$$\sigma_h(r) \;=\; 0.35\,\big(0.008 + 0.012\,r\big)\,\frac{H}{r}, \tag{4}$$

where r is the ray length to the cell, H the current camera-height estimate, and H/r the ray's vertical direction cosine—range error mostly slides a grazing ray *along* the floor, so only its vertical component (which shrinks with distance) corrupts height, attenuated further by per-cell averaging (factor 0.35). Two supporting fixes came from the same investigation: "flying pixels"—interpolated depth points along depth discontinuities—had been minting phantom mid-air cells and are now rejected at the sampler, and the hardcoded 1.4 m camera-height assumption was replaced by the continuous estimate described above.

5.6 Feedback design: pulse audio as a haptic-shoe prototype

Early versions spoke everything, which field walks quickly showed to be both too slow and too chatty: speech takes seconds to convey what a pattern conveys in a tenth of a second, and repeated announcements are abandoned by users [3, 11]. The final design (Table 2) reserves speech for nouns, route instructions, and urgent backstops, and delivers continuous guidance as generated stereo *pulse*

audio deliberately structured like a parking sensor—and deliberately structured to map one-to-one onto the planned haptic-shoe hardware (left/right actuation, pulse rate, intensity), so the eventual substitution of vibration for sound changes the transducer, not the code or the user's mental model.

Table 2: Pulse-audio encoding. Side maps to action, rate to distance, tone to hazard category. The same three channels map onto the planned haptic shoes (which foot, pulse rate, motor site).

Channel	Encodes	Scheme	Values
Side (stereo)	Action	left ear = move left; both = ahead	left / right / center
Rate	Distance	1.0 s interval at ≥2.5 m → 0.12 s at ≤0.5 m	continuous; urgent 0.10 s
Tone	Category	drop-off 220 Hz; obstacle 440 Hz; stairs/step 590 Hz	small step up 880 / down 660 Hz
Chime	All clear	single 1318 Hz chime, once	—

The pulse interval is a clamped linear map of hazard distance,

$$T(d) \;=\; T_{\min} + \big(T_{\max} - T_{\min}\big)\ \mathrm{clamp}\!\left(\frac{d - d_{\min}}{d_{\max} - d_{\min}},\ 0,\ 1\right), \tag{5}$$

with $(d_{\min}, d_{\max}) = (0.5, 2.5)$ m and $(T_{\min}, T_{\max}) = (0.12, 1.0)$ s, so perceived urgency grows continuously as distance closes—the same mapping a driver already knows from a parking sensor.

An *announcement governor* makes alerts trustworthy in time as well as in fact: a hazard is announced once per approach and re-announced only after the user closes 0.6 m or 10 s elapse (4 s for urgent classes); commands must persist in 8 of the last 12 frames (≈400 ms) before being committed; "path clear" requires 2 s of hazard silence. The 8-of-12 persistence gate is a probabilistic debouncer: under an idealized frame-independence assumption, a spurious detection appearing in any single frame with probability p issues a false command with probability

$$P_{\text{false}} \;=\; \sum_{k=8}^{12} \binom{12}{k} p^{k} (1-p)^{12-k}, \tag{6}$$

which is 5.8×10^{-4} even at a pessimistic $p = 0.2$ and 3.4×10^{-6} at $p = 0.1$—while delaying a *true* command by only ≈270 ms at 30 Hz. Real sensor errors are correlated across frames (a misread surface stays misread), which is precisely why this gate is complemented by the world-space hazard map's independent three-sighting rule and, for drop-offs, the physical vetoes of Section 5.8: the layers fail independently. These rates were tuned on field walks after early versions flip-flopped between commands at frame rate.

5.7 Performance: making it survive an iPhone 14 Pro

An early real-time build froze the target phone within half a second of launch. Diagnosis produced a chain of three distinct faults whose interaction is instructive. First, per-frame CPU-side image conversion (CIImage → CGImage → UIImage) allocated ≈11 MB and forced a GPU readback per 1920×1440 frame on the main thread; it was replaced by RealityKit rendering the camera feed directly on the GPU. Second, the pipeline ran synchronously on ARKit's delegate queue, so when a debug-build frame took 150–200 ms, ARKit retained a dozen in-flight frames, exhausted its fixed frame pool, and stopped the camera—the freeze. The fix is architectural: the delegate queue now does only constant-time work (hand the frame to an idle pipeline queue, or drop it), and at most one frame is ever in flight. Third, fixing the freeze *unmasked* a latent integer-overflow crash in drop-off promotion that had been unreachable while the app froze first—a reminder that in a failing system, later bugs hide behind earlier ones. Sustained-load work followed: per-cell statistics are recomputed only for cells touched by the current frame (eliminating ≈1,400 cells × two sorts per frame), ARKit mesh reconstruction was disabled after profiling showed it to be the dominant continuous background cost, and a thermal governor steps intake from 30 to 20 to 15 Hz at elevated thermal states while *stretching* hazard-persistence windows in wall-clock time, so throttling makes the system more conservative, never twitchier.

5.8 Truthful drop-offs: the war on the false urgent alert

"Drop ahead" is the system's highest-priority utterance (priority 100), so its false positives are the most corrosive. Field walks produced five distinct false-drop mechanisms, each requiring its own countermeasure:

1. **The LiDAR coverage boundary.** Floor returns die at 2–3 m under grazing incidence, so every column of the grid eventually reads "floor, then nothing"—indistinguishable, to naïve logic, from a cliff edge that moves with the user and re-announces forever. Fix: drop promotion is capped by the *measured* floor-coverage range (an EMA of how far floor was actually observed, minus a margin), not by field-of-view geometry.

2. **Discarded low-confidence depth.** ARKit marks grazing-range floor low-confidence even where depth is dense; the sampler had been discarding it, starving coverage. Fix: low-confidence samples may *reinforce floor only*, within ±0.12 m of the fitted plane—they can extend the ground, never mint a hazard.

3. **Shallow potholes.** A 12 cm pothole is a step-down, not a cliff. Fix: down-steps ≤20 cm whose

far side returns to floor within ≈45 cm are demoted to non-urgent small step-downs; a true drop must span ≥4 cells across ≥3 columns (a real edge crosses the corridor).

4. **Sensor blanks on dark or wet pavement.** LiDAR returns vanish on absorptive surfaces, mimicking missing floor. Fix: a *camera-luma veto* compares the candidate cell's image patch against confirmed-floor patches; same brightness and texture implies continuous pavement and vetoes promotion. Critically, the veto *fails toward warning*: at night, out of frame, or in replay (where no camera is recorded), it never suppresses.

5. **Sloped and crowned roads.** A single plane misreads legitimate road geometry as steps. Fix: a local quadratic ground surface

$$h(f, l) \;=\; c_0 + c_1 f + c_2 f^2 + c_3 l + c_4 l^2 \tag{7}$$

(forward f, lateral l) is fitted each frame to previously-committed floor cells by least squares over the basis $\boldsymbol{\phi}(f, l) = [\,1,\ f,\ f^2,\ l,\ l^2\,]$, solving the 5×5 normal equations $\mathbf{c} = (A^\top A)^{-1} A^\top \mathbf{h}$, followed by one trimmed refit that drops residuals exceeding $\max(3\hat{\sigma}, 2\,\text{cm})$ with $\hat{\sigma}$ from Eq. (2), and exponential smoothing of the coefficients ($\alpha = 0.2$). The correction is clamped to the steepest legal road grade (≈12%/m + 5 cm, ceiling 35 cm) so the model can never "absorb" a real stair rise.

The first countermeasure deserves its formal statement, because it inverts the usual reasoning: instead of asking "how far *could* the sensor see?" (field-of-view geometry), the system asks "how far *did* it see floor lately?". Per frame, the farthest positively observed floor distance r_k updates a coverage estimate

$$R_k \;=\; (1 - \alpha)\, R_{k-1} + \alpha\, r_k, \qquad \alpha = 0.05, \tag{8}$$

and missing depth may be interpreted as a drop *only* closer than $R_k - 0.5$ m—inside the range where the sensor has recently proven it can see floor when floor exists. Algorithm 1 gives the complete promotion logic; note that every gate that cannot be evaluated (no camera at night, no luma in replay, too few reference patches) defaults to *not* vetoing, so ambiguity always escalates toward the warning.

5.9 Stairs: detecting a structure, not a surface

The most consequential mid-project discovery was that the original stair logic was *dead code*: it required three step-classified cells per column, but the step class covers only 10–30 cm above floor, so on a real staircase (rises ≈17 cm) only the first step qualifies; steps two and beyond exceed 30 cm cumulative height and classify as obstacle. In practice a staircase read as "step up" + "wall ahead,

Algorithm 1 Drop-off promotion with fail-toward-warning gating (per frame, after classification passes A–B.2).

update floor-coverage EMA R_k by Eq. (8); $B \leftarrow \min(R_k - 0.5\,\text{m},\ \text{FOV band} - 0.3\,\text{m})$
for each grid column c not flagged stair-descending **do**
    $r_f \leftarrow$ last row of ≥ 2 consecutive confirmed-floor cells in c
    if rows $r_f{+}1 \ldots r_f{+}3$ are all unobserved for ≥ 0.09 s **and** lie within band B **then**
        mark cell $(r_f{+}1, c)$ a drop *candidate*
    end if
end for
cluster candidates (4-connected)
for each cluster X **do**
    if $|X| < 4$ **or** X spans < 3 columns **then**
        discard X ▷ potholes and occlusion shadows do not cross the corridor
    end if
    if camera luma available **and** mean luma ≥ 20 **and** ≥ 2 confirmed-floor reference patches exist **then**
        if $|\mu_X - \mu_{\text{floor}}| \leq 18$ **and** $\text{var}_X / \text{var}_{\text{floor}} \leq 3.0$ **then**
            veto X ▷ same brightness and texture $\Rightarrow$ continuous pavement
        end if
    end if ▷ gate unevaluable $\Rightarrow$ no veto: fail toward warning
    if X survives **and** intersects corridor ± 0.45 m **and** distance ≤ 2.5 m **then**
        emit DROPAHEAD (priority 100) through the governor
    end if
end for

turn left'—*the app actively steered users away from staircases*—and descending stairs read as an urgent drop. The replacement detects stairs as a *structure*: per-column height profiles are segmented into low-variance plateaus; a staircase is a run of ≥2 same-signed rises of 0.10–0.25 m (agreeing within 0.05 m) with tread spacing 0.20–0.45 m, coherent across ≥4 adjacent columns (Algorithm 2). Ramps produce no plateaus and correctly do not match; a single curb produces one rise and correctly stays a step. Descending-stair columns are exempted from drop promotion.

Algorithm 2 Structural stair detection from column height plateaus.

for each grid column c **do**
  $h_r \leftarrow$ baseline-corrected cell heights (ground model, Eq. (7))
  segment h_r into *plateaus*: maximal runs of ≥2 rows agreeing within 3 cm, merging occlusion gaps of ≤3 rows
  $\Delta_j \leftarrow$ height difference between consecutive plateaus $j, j{+}1$ ▷ the floor is plateau zero
  find the longest run of same-signed rises with $0.10 \le |\Delta_j| \le 0.25$ m, pairwise agreement ≤ 0.05 m, and tread spacing 0.20–0.45 m
  if run contains ≥ 2 rises **then** mark c *stair-like* with its first-step row and rise sign
  end if
end for
cluster stair-like columns: ≥ 4 adjacent columns whose first-step rows drift by ≤ 1 row ▷ a staircase is laterally coherent; clutter is not
for each cluster **do**
  override member cells' OBSTACLE labels ▷ steps 2+ exceed the 30 cm obstacle threshold
  sign of median rise ⇒ STAIRSUP / STAIRSDOWN; exempt columns from Algorithm 1
end for

A four-state climb guide (idle→approaching→climbing→top reached), driven by the camera's world height, speaks approach and alignment cues, mutes obstacle chatter during the climb (the staircase itself would otherwise announce as walls), and keeps only *stop* and *drop* live. Counting individual steps was explicitly rejected in the design spec: the user also carries a cane, and the cane is better at step edges than any sensor—the system's job is finding the staircase, aligning to it, and knowing when it ends.

5.10 People are not terrain

During a field walk, a pedestrian crossed the corridor left-to-right, and PathSense computed the geometrically correct sidestep—*directly into the pedestrian's path*. Static-world logic treats a person as a fixed obstacle to route around; a person is a moving agent to yield to. The fix promotes people to first-class tracked objects: detections are ray-cast at their footpoint onto the floor plane (working beyond LiDAR range), tracked with velocity estimates, and classified as stationary, crossing, approaching, or receding. Three behavioral rules replaced geometry: never steer in front of a

crossing person (wait if they clear the corridor within 2.5 s; otherwise pass *behind* them, on the side they came from); a person walking away is a *leader* to follow silently, alerted only if the user closes on them within 2.2 m (spoken, not pulsed—a centered obstacle pulse would invite exactly the sidestep the rule exists to prevent); and moving-person cells are retyped before world-map ingest so passersby stop minting 25-second phantom obstacle memories. A second failure a day later—a person walking away read as stationary terrain, producing a trail of phantom obstacles—added the receding class with speed hysteresis.

5.11 The mirrored-steering defect: why conventions need tests

The single most dangerous defect in the project's history was also its simplest: the grid's rightward basis vector was constructed as $up \times forward$, which points *left*. Every lateral semantic in the system inherited it end to end—sidestep directions, stop-versus-pivot turns, stereo pan, the minimap. A user with clear space on their right was told "move left." The bug was caught the day the project's first unit tests were written, when a beacon-geometry test disagreed with the grid's convention, and fixed by a one-line change ($forward \times up$); the sign convention is now pinned by tests in four independent subsystems. The lesson generalizes: in safety systems, *coordinate conventions are load-bearing claims and must be tested like any other claim*—visual inspection had passed the mirrored build repeatedly because debug overlays mirrored consistently.

5.12 Navigation without a compass

Turn-by-turn guidance needs the user's true heading, but a chest mount's magnets make the magnetometer unusable. PathSense recovers heading by fusion: over a sliding window, it compares GPS course-over-ground against the AR session's displacement bearing and estimates the constant offset between ARKit's drift-corrected yaw and true north, $\theta_{\text{true}} = \theta_{\text{AR}} + \hat{\delta}$. Because headings are angles, naïve averaging fails at the $359^\circ/1^\circ$ wraparound; each observed offset δ_i is therefore embedded on the unit circle and smoothed as a vector,

$$\mathbf{v}_k \;=\; (1-\alpha)\,\mathbf{v}_{k-1} + \alpha \begin{bmatrix} \cos\delta_k \\ \sin\delta_k \end{bmatrix}, \qquad \hat{\delta} = \operatorname{atan2}(v_y, v_x), \qquad \alpha = 0.2, \tag{9}$$

the circular-mean construction of directional statistics [17]. The resultant length $\|\mathbf{v}_k\|$ doubles as a self-consistency score—mutually agreeing samples push it toward 1, contradictory samples cancel toward 0—and is published as the fusion confidence, $\min(\|\mathbf{v}_k\|/0.6,\ 1)$. Samples are admitted only under strict gates (speed $\geq$0.7 m/s, straight-line agreement across window quarters within 20°, GPS course accuracy $\leq$30°, displacement $\geq$1.5 m), which reject mid-turn and GPS-multipath geometry.

Route cues then ride the smooth 30 Hz AR yaw rather than raw GPS. The route beacon—a spatial audio target with direction triple-encoded in timbre, exaggerated stereo pan, and tempo—arms only when fusion confidence exceeds 0.5, encoding a core principle: *a wrong direction is worse than none*. A clock-face environment scan (five sectors, 10–2 o'clock, three rays each) speaks a summary such as "clear ahead 4 meters; wall on your right" when the user stands still, claiming "clear" only where floor was positively observed—absence of data is never evidence of absence of hazard.

5.13 Intersection safety as an evidence-gated state machine

The route and obstacle systems can assist near an intersection, but a traffic-light color is not a crossing decision. The default-off crossing research subsystem therefore separates pedestrian-signal state from vehicle-signal state at the type level: no vehicle red, yellow, or green value can permit or prohibit pedestrian entry. The only potentially authorizing signal value is a pedestrian WALK observation, and even it remains one required witness rather than a command. Signal history distinguishes a WALK whose onset was observed from one already illuminated when observation began, because the latter has unknown remaining time. This design follows the MUTCD distinction between pedestrian WALK and flashing/steady upraised-hand indications and PROWAG's association of accessible walk indications with a particular crosswalk [18, 19].

Let C mean that the intended crosswalk and route bearing are associated, W that a newly observed pedestrian WALK indication is stable and fresh, S that the required left–forward–right traffic scan is complete and fresh, V that no tracked or occluded vehicle conflict is present, H that the crossing-specific sensing capabilities are healthy, and B that local terrain and body-space evidence contain no hard hazard. A necessary research gate for any entry-related advisory is

$$G_{\text{entry}} = C \wedge W \wedge S \wedge V \wedge H \wedge B. \tag{10}$$

If any term is false or unknown, the only permitted result is WAIT, HOLD, or "crossing assistance unavailable." Even when all terms are true, the research wording is informational—for example, "pedestrian WALK signal detected"—rather than "safe to cross." Turning vehicles, occluded lanes, and stale scan coverage remain physical threats.

The implemented shadow state machine distinguishes approaching, locating the curb, waiting, checking traffic, entering, crossing, reaching the far curb, and completion (Figure 1). LiDAR down-step geometry proposes near-curb lines, route context proposes an intended bearing, dynamic tracks provide time-to-conflict intervals, scan coverage records observed and occluded approach sectors, and an observed up-step line is required for the far curb. Ambiguous signal heads, missing route bearing, stale scans, or an unobserved far curb remain explicit refusals. The before/after-entry

asymmetry prevents a stateless response: signal loss before entry yields a wait, while signal loss after entry does not by itself command a stop in a traffic lane or restore ordinary navigation.

This subsystem is E2 shadow software, not crossing assistance. Its feature flag defaults off, static checks prevent it from reaching the guidance bus, and no validated pedestrian-signal, crosswalk-marking, curb-ramp, or roadway model is bundled. The entry gate therefore remains closed independently of the behavioral policy. Construction barriers and confirmed sidewalk closures are still unimplemented as persistent no-go polygons and reroute requests; they remain an important E0 gap.

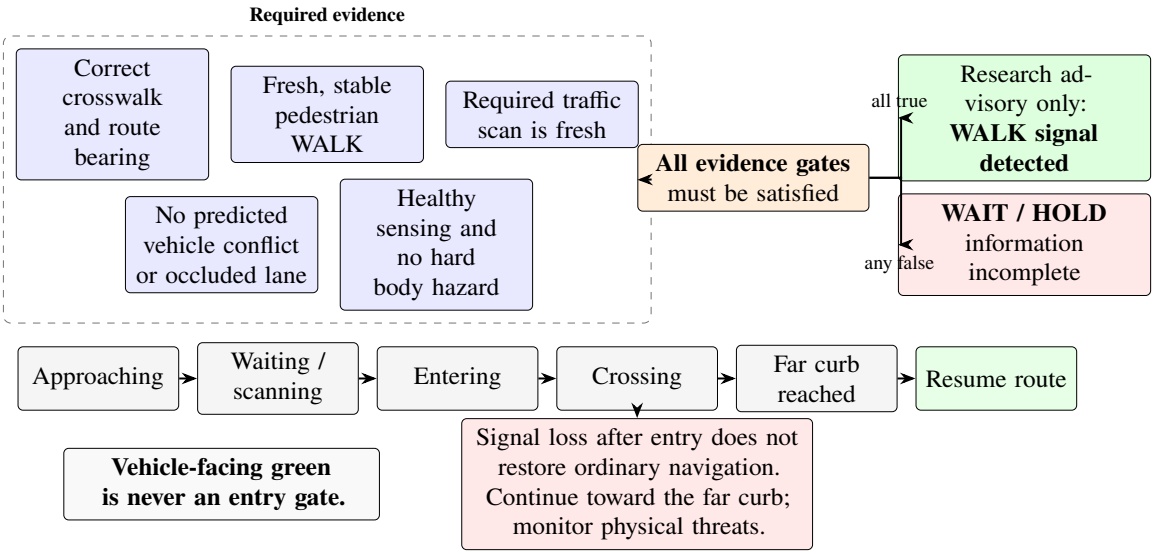


Figure 1: Implemented shadow intersection logic. Every entry gate must be supported; vehicle-facing green is deliberately excluded. After entry, physical-threat monitoring and far-curb progress remain active even if the signal changes. The diagram does not represent enabled crossing guidance.

5.14 The semantic object layer: labels annotate, never veto

Before the semantic layer, the system was geometry-only: obstacles were anonymous height blobs ("obstacle ahead, 8 feet"), perception ended at the 2–3 m LiDAR floor limit, and person detection ran a slow 5 Hz heuristic. The design spec for the semantic layer records why the obvious alternatives were rejected: ARKit mesh classification had already been removed for thermal cost and is unreliable outdoors, and a GPU detector would compete with rendering. The enabling observation was architectural: the Apple Neural Engine is *separate silicon*, so a small detector runs there almost free of the CPU/GPU thermal budget that had nearly killed the project in Section 5.7. YOLO11n runs at 10 Hz (degrading to 5/3 Hz under thermal pressure), detecting a 13-class COCO subset relevant to sidewalks (person, bicycle, car, bus, truck, dog, bench, chair, fire hydrant, . . .).

Fusion with geometry is deliberately asymmetric, expressed as the project's first safety invariant:

> **Invariant 1 (camera semantics never clear LiDAR occupancy).** *A semantic label may name an occupied region or add conservative cost, but it may not convert LiDAR-occupied evidence into free space or extend the lifetime of stale geometric evidence.*

Let $O_L(x,t)$ be LiDAR occupancy risk and let $O_F(x,t;S)$ be fused occupancy in the presence of semantic observations S. Under the ordered risk scale free < unknown < occupied, the invariant is

$$O_F(x,t;S) \;\succeq\; O_L(x,t) \quad \text{whenever } O_L(x,t) = \text{occupied}. \tag{11}$$

Thus semantics can change a noun, add a finite comfort cost, or convert a sidestep into a stop, but cannot erase metric obstruction evidence. The separate camera-luma drop veto is not a semantic-label override: it is a photometric continuity test with strict availability gates, and it defaults to no veto when its evidence is unavailable.

A detection is localized by sampling the depth map on an 11×11 lattice over the central $60\% \times 50\%$ of its box and taking the median,

$$\hat{d} = \text{median}\{\, d(x) : x \in \text{box core},\ \text{conf}(x) \geq \text{medium} \,\}, \tag{12}$$

accepted only with ≥12 valid samples and $\hat{d} \leq 4.5$ m (the median again buys outlier robustness: a box half-filled with background depth still localizes on the object). Beyond LiDAR, the box's footpoint is ray-cast onto the floor plane to ≈15 m at "awareness grade." Physical extent follows from the pinhole model, $w = w_{\text{px}}\,\hat{d}/f_x$, giving the planner a real-world footprint. Tracks accumulate label votes—the committed label is the mode over the last eight detections, committed only after three hits—so a detector flickering between "car" and "truck" at 10 Hz cannot flicker speech. The user-facing result is precision without new risk: the same governed announcement now says "*Car* ahead, 8 feet," and a separate long-range awareness channel announces vehicles, bikes, and dogs at 3.5–12 m once per approach. Two hallucination defenses came from field walks: below a luma floor the detector stands down entirely (a COCO model on night-noise frames hallucinates), and in a dusk band it runs with raised confidence thresholds *plus* required LiDAR corroboration—a hallucinated box has no coherent depth behind it. Semantic tracks do influence *comfort* routing, but only through the second invariant: the planner's cost field may be modified additively and finitely,

$$c'(x) \;=\; c(x) \;+\; \sum_i \kappa_i\, \mathbf{1}\big[x \in \text{disc}_i\big], \qquad 0 \leq \kappa_i < \infty, \tag{13}$$

(a labeled car stamps a disc of $\kappa = 20$; a moving track adds a half-cost disc at its velocity-

predicted position). Because every κ_i is non-negative and finite, a label can make a route *less attractive* but can never erase an infinite-cost geometric hazard or manufacture a shortcut. Likewise, the semantic sidestep veto is a monotone map on the command lattice—it may only send $\{\textsc{moveLeft}, \textsc{moveRight}\} \mapsto \textsc{stop}$, never the reverse—and it activates only on measurement-grade (LiDAR-confirmed, ≥5-hit) tracks: strictly more conservative, never less.

One practical failure is worth recording for reproducibility: exporting YOLO11n at the camera's native 640×480 aspect crashes the Core ML converter on the model's attention block (verified on two toolchain versions); the export therefore uses a square 640 input with scale-to-fill, accepting a known ≈33% aspect distortion—the label voter and confidence gates absorb the accuracy cost, and the toolchain pin is documented so the result is reproducible.

5.15 GuideSpace: vertical evidence and body-relative risk

The active planner still consumes the established 7.5 cm terrain grid. In parallel, the newest research layer runs a shadow representation called *GuideSpace*. Its purpose is to preserve information that a single top-down class cannot express: whether an obstacle threatens the feet, lower legs, torso, or head; whether space is occupied, observed free, or unknown; which sensor supplied the evidence; and how old that evidence is. Because GuideSpace is shadow-only, the results in this paper do not attribute live guidance benefit to it.

GuideSpace divides the horizontal world into 10 cm columns. Each column contains 30 vertical bins of height 10 cm spanning approximately −0.4 to 2.6 m relative to the local walking surface (Figure 3). Full-confidence LiDAR hits add normal occupancy evidence. Low-confidence hits add half-weight evidence and are capped below confirmed occupancy so weak returns cannot dominate strong geometry. Only full-confidence hit rays may reduce stale occupancy along observed free space. A free-space ray never creates a column, stops before its measured surface, and applies at most one multiplicative decay to a bin per frame. These constraints prevent repeated rays or missing returns from manufacturing certainty.

Camera labels attach only to bins that already contain LiDAR occupancy. Conflicting labels may coexist with confidence weights, but labels neither change occupied/free evidence nor refresh geometric retention. The resulting authority hierarchy is shown in Figure 4: metric hits may establish occupancy, high-confidence rays may age it away, and camera semantics may answer "what is it?" but not "is the path clear?".

A *TravelerProfile* projects the vertical evidence into ground, lower-leg, torso, and head risk slices. Each slice stores occupancy risk, unknown risk, minimum clearance, age, evidence source, and semantic class. The default profile uses height 1.75 m, body radius 0.35 m, walking speed 1.2 m/s,

and reaction time 0.75 s. These are configurable engineering defaults, not population estimates. The initial stopping-distance model is

$$d_{\text{stop}} \;=\; v\,t_r + 0.5\ \text{m}, \tag{14}$$

where v is current walking speed and t_r is the reaction-time allowance. Figure 2 shows the sensitivity of this engineering model over a plausible design range. It is not a measured distribution of participant reaction time or walking speed.

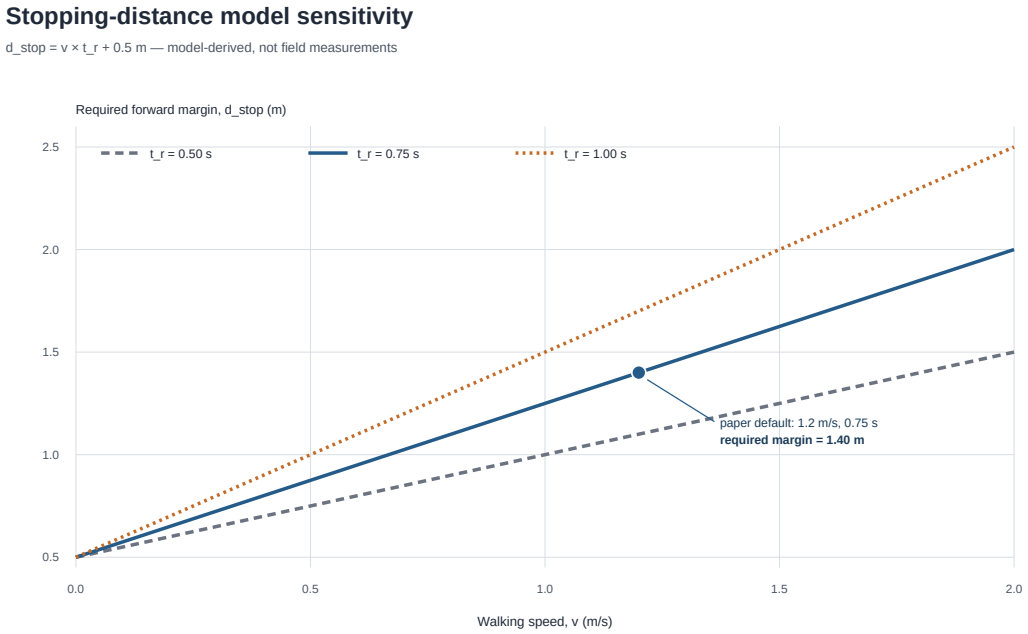


Figure 2: Model-derived stopping-distance sensitivity. The highlighted point is the paper's configurable default profile; the curves are deterministic outputs of Equation 14, not field measurements.

GuideSpace now feeds a default-off, deterministic *Conformal Action–Path Envelope* (CAPE) shadow evaluator. CAPE generates a bounded action vocabulary, sweeps the configured traveler volume through all body bands, charges unknown and stale evidence, and applies dynamic and topological rejection before ranking surviving candidates. Its intended calibrated criterion remains a positive lower clearance margin,

$$m_{\text{lower}}(\gamma) = \min_{s\in\gamma}\left[c(s) - r_{\text{body}} - q_{1-\alpha}(s)\right] > 0, \tag{15}$$

where $c(s)$ is predicted clearance and $q_{1-\alpha}$ is a held-out, route-separated calibration allowance for perception and localization error. The current evaluator implements the geometric and freshness gates but does not possess a participant-derived conformal allowance, so Equation 15 remains a calibration target rather than a statistical guarantee. CAPE publishes an expiring action envelope,

candidate scores, rejections, minimum clearance, capability dependencies, and computation cost for replay. It remains shadow-only. A separate Research-Mode caution ratchet can only escalate an existing command to slow or stop and also defaults off; it cannot steer or clear a legacy hard stop.

The same scene also feeds a curb-latched topological safety field. Geometric emptiness and pedestrian admissibility are separate axes: a roadway can be empty and still forbidden to ordinary travel. Curb boundaries retain sidedness across short occlusions, stale positive permission decays to unknown, and a stale prohibition does not silently open. CAPE applies this field as an irreversible rejection after the body sweep and before dynamic prediction. Route confidence and camera semantics cannot revive a rejected road-side detour. Figure 5 keeps these evidence levels visible so test-backed shadow software is not mistaken for deployed behavior.

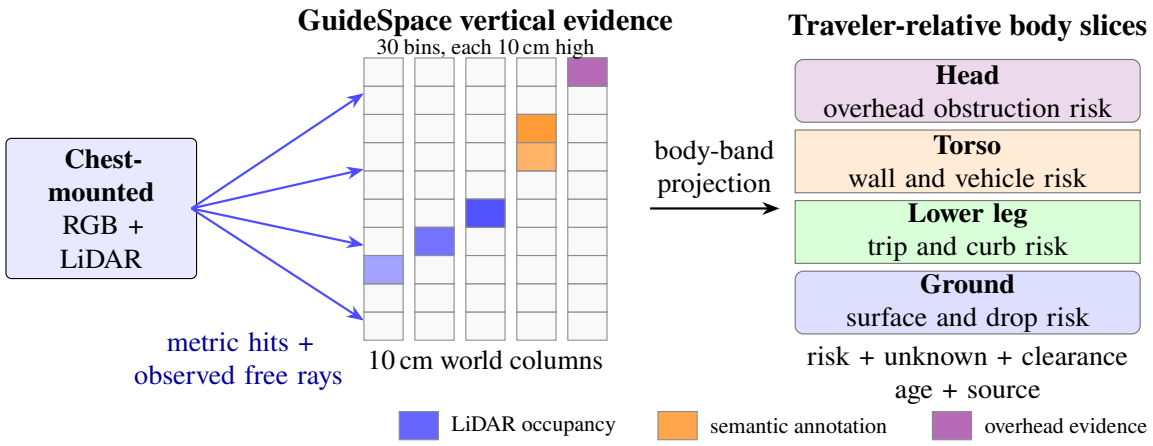


Figure 3: GuideSpace preserves vertical occupancy and unknown evidence, then projects it into traveler-relative risk bands. CAPE consumes these slices in shadow; the active planner still uses the established terrain grid.

5.16 Coordinated research stack: route intent to one action

The integrated research stack treats navigation as a hierarchy rather than a set of independent features (Figure 6). The global route supplies a destination preference; RouteLock provides a conventional map-matching baseline; the *Perceptual Route Weave* (PRW) retains multiple route-edge and pedestrian-region hypotheses; GuideSpace, body slices, dynamic tubes, and the topological field constrain local motion; CAPE evaluates a bounded candidate set; and guidance-response models ask what motion a cue is likely to produce. Every component publishes a typed result or a typed refusal through one versioned snapshot. The integrated coordinator is structurally isolated from the traveler-facing bus, so route progress, route memory, or camera evidence cannot override a geometric stop, topological rejection, or unhealthy capability.

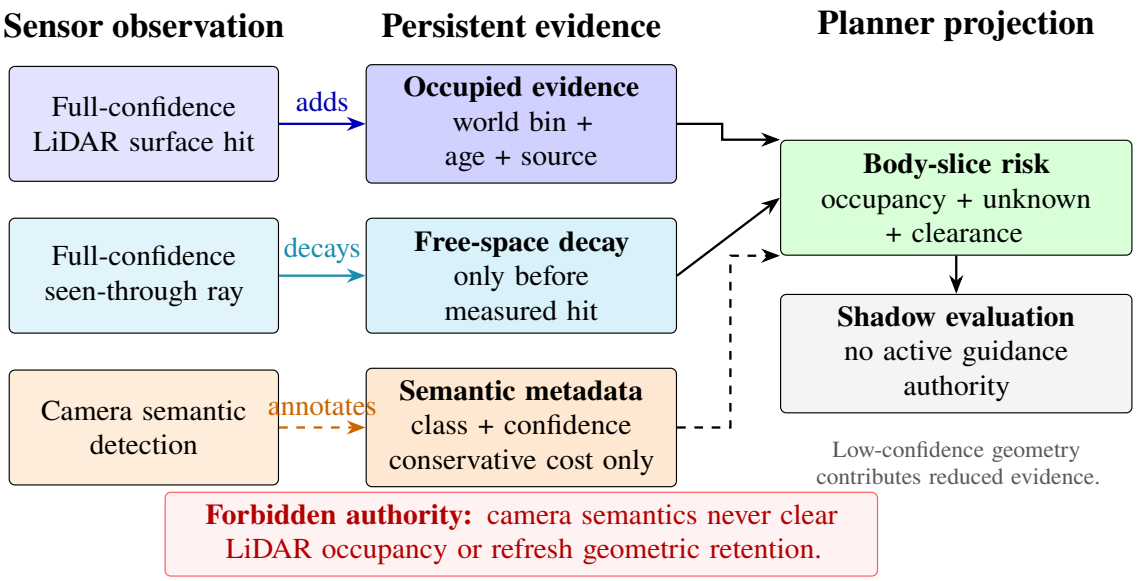


Figure 4: Asymmetric evidence authority. Solid arrows change geometric evidence; dashed arrows add metadata or conservative cost only.

System or claim	Runtime status	Strongest evidence
Terrain, local A*, route guidance, semantics, and output bus	Active / gated	E2 + selected E3
GuideSpace, CAPE, dynamic tubes, and curb-latched topology	Shadow	E2
Route graph, PRW belief, route intent, and counterfactual scans	Shadow	E2
G-CAPE, maneuver witnesses, attention policy, and route memory	Implemented / limited	E2 synthetic
Crossing lifecycle and conflict fusion	Shadow; models absent	E2 synthetic
Construction closures, GaitLock, and temporal evidential model	Proposed / absent	E0
Blind or low-vision participant benefit	Not evaluated	No E4 evidence

Figure 5: Research status and strongest evidence presently available. E0: proposal; E1: code/design review; E2: build, unit, synthetic, or replay verification; E3: developer physical-device observation; E4: supervised participant evidence.

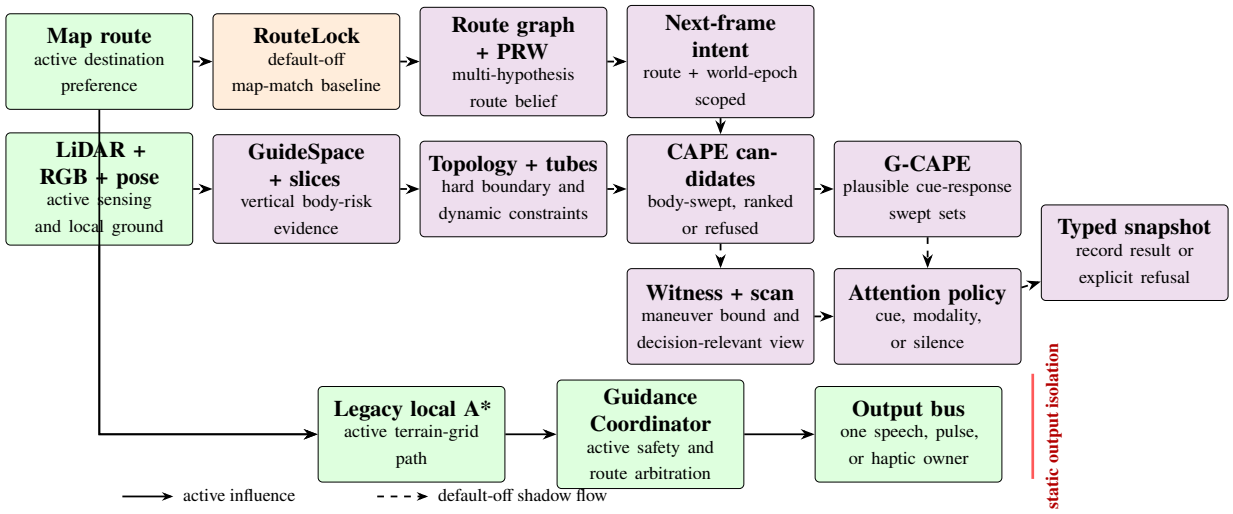


Figure 6: Active and research-only authority paths. Solid arrows presently influence traveler guidance; dashed arrows remain default-off shadow comparisons. The legacy A* planner remains the active local planner.

Candidate trajectories and lexicographic authority. The shadow generator produces stop, straight, reduced-speed, gradual and strong left/right, heading-hold, RouteLock-aligned, and PRW-aligned trajectories. Each records route progress, static clearance, ground and unknown exposure, topological exposure, dynamic conflict, turn complexity, path-change cost, and an optional semantic explanation. A representative deterministic score is

$$\begin{aligned} J(\gamma) = {} & w_s R_{\text{static}}(\gamma) + w_d R_{\text{drop}}(\gamma) + w_m R_{\text{motion}}(\gamma) + w_u U(\gamma) \\ & + w_r D_{\text{route}}(\gamma) + w_\theta C_{\text{turn}}(\gamma) + w_h D_{\text{previous}}(\gamma), \end{aligned} \tag{16}$$

for candidate path γ. The implementation does not use this sum to buy its way through hard constraints: body collision, drop, topological boundary, controlled crossing, and blocking dynamic conflict reject the candidate before score comparison. The accessible route graph applies the same principle lexicographically by removing forbidden edges before comparing hazard bound, uncertainty, crossing burden, instruction burden, and time. Thus the route can rank already-admissible choices but cannot make an unacceptable one traversable.

Clearance inflation and CAPE. The deterministic shadow evaluator inflates clearance using body size, tracking state, speed, and evidence age. The intended calibrated form is

$$r_{\text{req}}(t) = r_{\text{body}} + \epsilon_{\text{pose}} + \epsilon_{\text{ground}} + \epsilon_{\text{motion}}(t) + v\,\tau_{\text{age}} + q_{1-\alpha}(z), \tag{17}$$

where τ_{age} includes sensing and processing age and $q_{1-\alpha}(z)$ is a calibration allowance for context z. CAPE accepts only if $\min_t [c_\gamma(t) - r_{\text{req}}(t)] > 0$ and all required capabilities are healthy. The current `PathRiskEvaluator` performs the deterministic body sweep, rejects stale or missing evidence, and records an expiring envelope. No real calibration artifact is bundled, so the $q_{1-\alpha}$ term and any claimed conformal coverage remain unavailable.

RouteLock and Perceptual Route Weave. RouteLock is a default-off map-matching baseline that reports route progress, cross-track error, bearing, confidence, and named refusal reasons. PRW extends that baseline by maintaining a normalized belief over route edges and observed off-route regions, $h_i = (e_i, s_i, \ell_i, \psi_i)$, where e_i is an edge or unmapped region, s_i is progress, ℓ_i is pedestrian-region identity, and ψ_i is world-to-map yaw. In log space, with each L denoting a capped log-likelihood, its update is

$$\begin{aligned} b_t(h_i) \propto\ & L_{\text{GPS}}(h_i) + L_{\text{VIO}}(h_i) + L_{\text{heading}}(h_i) \\ & + L_{\text{topology}}(h_i) + L_{\text{sequence}}(h_i) + \log \sum_j \exp\left[\log p(h_i \mid h_j, u_t) + b_{t-1}(h_j)\right], \end{aligned} \tag{18}$$

written in log space. Likelihood-family caps prevent correlated evidence from creating false certainty, and RouteLock itself is recorded only as a baseline because scoring it together with its raw GPS and heading inputs would double count evidence. PRW publishes a route intent only when the best hypothesis exceeds a lock threshold, dominates the runner-up, has bounded entropy, is fresh, and agrees with topology. An unmapped-region hypothesis may motivate a scan but cannot create a route candidate. The intent is timestamped, scoped to the route fingerprint and AR-world epoch, and consumed only by a later frame, so same-frame causality is never implied.

Uncertainty-expanded dynamic tubes. The default-off shadow predictor converts person, bicycle, vehicle, and unknown-mover tracks into time-indexed occupancy tubes. Each tube separates an observed-motion core from a maneuver halo; stale and occluded tracks coast briefly while uncertainty grows, then expire by class. For object k and motion hypothesis j,

$$\mathcal{T}_{k,j}(t) = B\left(\hat{\mathbf{p}}_{k,j}(t),\ r_k + \beta\sqrt{\lambda_{\max}(\Sigma_{k,j}(t))}\right), \tag{19}$$

where $B(\mathbf{p}, r)$ is a ground-plane disc, Σ is predicted position covariance, and the radius expands with uncertainty and horizon. A candidate is rejected when its time-aligned body sweep reaches the core before the traveler can stop; halo-only overlap constrains speed and adds risk. Dynamic prediction is monotone: it may reject or constrain, but cannot clear static occupancy or unknown space. The current rule-based tubes are E2 shadow software, not a learned human-intent model.

Guided CAPE: instructions as uncertain actuators. Ordinary CAPE evaluates an ideal geometric trajectory. Guided CAPE (G-CAPE) instead evaluates the set of motions a traveler might execute after a cue g. The time-indexed swept response set is

$$\mathcal{T}_g(t) = \hat{x}_g(t) \oplus \mathcal{B}_{\text{body}} \oplus Q_{1-\alpha_u}(\epsilon_{\text{user}}) \oplus Q_{1-\alpha_p}(\epsilon_{\text{perception}}) \oplus Q_{1-\alpha_l}(\epsilon_{\text{localization}}), \tag{20}$$

where $\oplus$ is the Minkowski sum and the Q terms are separately reportable error regions. The implemented deterministic response model includes cue latency, heading response, drift, overshoot, missed cues, and continuation of the prior action. The guided sweep rejects any response mode that reaches occupied, unknown, topologically forbidden, crossing-controlled, or dynamically conflicting space. The evaluator and calibration-artifact validator are E2 software, but no real artifact is bundled and the live frame path does not yet drive G-CAPE; its truthful operational state is therefore uncalibrated or unavailable, not a conformal guarantee.

Witness-bounded maneuver commitment. A turn should not commit merely because several correlated confidence values look persuasive. Given lower bounds L_i for independent required witness *claims*—correct approach, expected opening, pedestrian-region continuity, heading change, and post-turn confirmation—the maneuver machine uses the Fréchet/union lower bound

$$L_{\cap} = \max\left(0,\ 1 - \sum_{i=1}^{n}(1 - L_i)\right). \tag{21}$$

The state machine and bound are implemented and tested with synthetic witnesses, including crossing restrictions, but live opening and boundary-orientation witness producers do not exist. It therefore refuses live maneuver commitment rather than treating absent witnesses as success.

Counterfactual active scanning. The scan planner optimizes uncertainty that can change the current decision, not the largest unknown image area. For bounded scan action a,

$$U(a) = \frac{H_D(b) - \mathbb{E}_z[H_D(b^{a,z})]}{T(a) + \lambda_A A(a)} - \lambda_E E(a), \tag{22}$$

where H_D is entropy over relevant actions, T is scan time, A is attention/motion cost, and E is exposure while scanning. The default-off planner can recommend left/right, downward-boundary, forward-structure, wait, or no-scan actions and records whether a later view was actually visited before judging the recommendation. It never prompts the traveler in the current build.

Attention-budgeted guidance and route memory. After unsafe cue executions are removed, the shadow attention policy ranks speech, spatial audio, haptic, and silence candidates by information value, attention cost, repetition, and reversal. Emergency and health cues short-circuit the attention budget, so a stop cannot be silenced because the traveler has recently heard many instructions. Silence is an explicit candidate, not merely failure to choose. This policy is implemented and tested but is not driven from the live frame path.

A separate consented route-memory store retains bounded structural/topometric descriptors, provenance, age, and change state locally. Memory may influence only a capped PRW likelihood or scan expectation; it cannot mark current occupancy clear. Revocation prevents writes and deletes the local store idempotently. No route-memory upload path exists, and no repeated-route benefit is claimed.

Gait-aware sampling and temporal evidence. GaitLock remains a separate E0 experiment: estimate gait phase from IMU and camera motion, transform measurements toward a stable reference phase, and give low-angular-velocity observations more geometric authority. A simple sampling weight is

$$w_g(t) = \exp\left(-\frac{\|\boldsymbol{\omega}(t)\|^2}{2\sigma_\omega^2}\right)\exp\left(-\frac{|a_y(t)-g|^2}{2\sigma_a^2}\right). \tag{23}$$

A later causal evidential model could update nonnegative evidence $\mathbf{e}_t = F_\theta(\mathbf{e}_{t-1}, z_t)$ for floor, obstacle, drop, stairs, and future occupancy. With $\alpha_{t,k} = e_{t,k} + 1$, total $S_t = \sum_k \alpha_{t,k}$, and K classes, the explicit unknown mass $u_t = K/S_t$ remains high when evidence is weak. Learned output would initially be authorized only to add cost or request another observation, never to clear LiDAR occupancy.

Recorded digital-twin simulation. ClipV2 now records and decodes synchronized RGB, depth, confidence, pose, motion, route, health, guidance, events, rendered-cue responses, and typed perceptual-navigation snapshots. Frame identity is authoritative: a missing sample is refused rather than shifting later streams by array index. Subsystem-scoped evidence gates require complete, compatible coverage before a metric can use a clip. An offline reconstruction can later act as a higher-information teacher, separate static structure from tracked actors, and perturb mount angle, speed, reaction delay, GPS/heading error, sensor confidence, processing age, or traveler dimensions. For one frozen episode s and perturbation θ, paired comparison reports

$$\Delta M(s,\theta) = M_{\text{candidate}}(s,\theta) - M_{\text{baseline}}(s,\theta), \tag{24}$$

for clearance, unknown exposure, stops, reversals, or conflicts. This can reject weak algorithms and discover failure regimes, but it cannot establish human safety: disagreement is not ground truth, labels remain necessary, and a changed live action would change subsequent observations.

5.17 Local path planning and the planner view

For situational awareness (and for sighted orientation-and-mobility trainers walking alongside), PathSense renders a Tesla-FSD-style planner scene built on an A* search [20] over the terrain grid. The traversal cost of a cell composes three terms,

$$c(x) \;=\; c_{\text{class}}(x) \;+\; c_{\text{infl}}(x) \;+\; c_{\text{sem}}(x), \tag{25}$$

where c_{class} encodes terrain (floor 1.0, unknown 1.6, bump 2.5, small step 6, step 20, stairs 30, obstacle and remembered drops ∞), c_{infl} inflates obstacles by a hard 0.35 m shoulder radius plus a soft penalty ring to 1.5×, and c_{sem} is the additive semantic term of Eq. (13). The search minimizes $f(n) = g(n) + h(n)$ with the Euclidean heuristic $h(n) = \lambda \, \|n - \text{goal}\|_2$ scaled by the minimum per-meter cost λ; since no traversal costs less than λ per meter, h never overestimates, so it is admissible and the returned path is optimal in the cost field [20]. Path stability is enforced by hysteresis—cells within one cell of the previous path are discounted ×0.8, and a new route replaces the old only if it is at least 15% cheaper—so the rendered path does not flicker between equal-cost alternatives frame to frame. Obstacle cells are clustered into recognizable objects—walls, fences, poles, blocks—by PCA-based geometric segmentation with a LiDAR *permeability* test distinguishing fences (LiDAR passes through gaps) from walls, corroborated by ARKit vertical planes. An Apple Maps drape underneath fades in only when heading-fusion confidence is high (an unseeded transform shows *no* map rather than a wrong one). The planner is display-only by design: audio guidance never reads it, so a rendering bug cannot steer a user.

6 System Architecture and Experimental Layers

Figure 7 shows the active data flow and research branch; Figure 8 shows the sensing geometry; Table 3 lists principal engineering parameters. Three properties merit emphasis.

Two-queue real-time core. ARKit's delegate queue performs only constant-time intake (claim the idle pipeline or drop the frame); a serial pipeline queue processes at most one frame at a time through sampler → floor tracker → ground model → grid → classifier → hazard map → stair guide → decision. Results cross to the main actor in strict FIFO order so command hysteresis sees every

frame exactly once.

Priority-arbitrated single voice. Every audio-capable subsystem—safety commands, route guidance, beacon, environment scan, semantic awareness—submits to one arbiter; exactly one command owns the audio channel, chosen by fixed priority, and every lower layer yields. The user never hears two systems talk over each other.

Explicit unknown and health-aware fallback. Unknown cells are never called "clear"; the beacon silences below localization confidence; and stale processing, mount uncertainty, or insufficient evidence should produce a slow, stop, or scan state. This is more precise than a blanket "fail toward warning" rule: inventing an obstacle from absent depth is also unsafe because it can mask a real route or cause alarm fatigue.

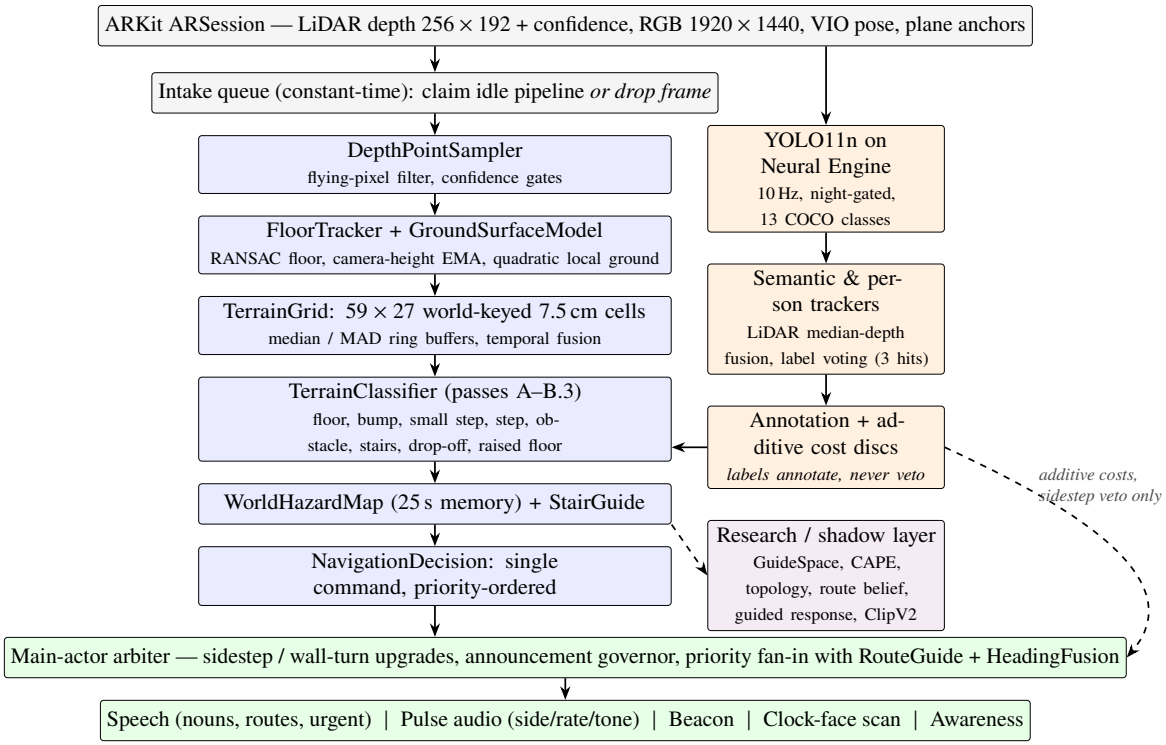


Figure 7: PathSense data flow. Blue: the active serial geometry pipeline. Orange: the semantic layer; its outputs may annotate occupied evidence and add conservative cost but cannot clear LiDAR occupancy. Green: single-voice arbitrated outputs. Violet: shadow world modeling and research navigation, all statically isolated from traveler-facing output.

7 Failure Catalog

Table 4 condenses the project's documented failure–diagnosis–fix cycles and later safety audits. Rows marked † have dedicated automated regression coverage; open audit findings remain labeled

Table 3: Principal engineering parameters of the active and shadow systems. Values are implementation settings requiring sensitivity analysis, not population-level constants.

Parameter	Value	Rationale / origin
Sensing and grid		
Grid cell size	7.5 cm	half a foot-length; resolves curbs
Grid extent	-0.4 to 4.0 m $\times$ $\pm$1.0 m	LiDAR useful range
Depth stride / samples	every 4th px ($\approx$3,000/frame)	CPU budget
Depth noise model	$\sigma(d) = 0.008 + 0.012\,d$ (m)	empirical fit
Pipeline rate	30 / 20 / 15 Hz	thermal tiers
Classification		
Floor tolerance	$\pm$8 cm	outdoor pavement variation
Small-step edge	max(4 cm, 4 $\times$ MAD), $\leq$2 m	noise-adaptive (Sec. 5.5)
Step-up band	10–30 cm	curb heights
Obstacle threshold	$\geq$30 cm, 5 frames	above-step structures
Drop-off min depth	15 cm (raised from 12)	pothole field tuning
Stair rises	0.10–0.25 m, $\pm$0.05 m agreement	building code range
Stair treads	0.20–0.45 m spacing	building code range
Hazard memory	25 s, $\geq$3 sightings	passing-hazard recall
Guidance		
Stop / slow bands	0.7/0.95 m; 1.3/1.7 m (enter/exit)	hysteresis
Announce distances	hazards 2.5 m; stairs 3.5 m	reaction time at walking pace
Command persistence	8 of 12 frames + 3 s dwell	anti-flip-flop
Re-announce	after 0.6 m closed or 10 s (4 s urgent)	governor
Semantic layer		
Detector rate	10 / 5 / 3 Hz (thermal)	$\approx$10% ANE duty
Confidence gates	0.45–0.55 by class; +0.15 at dusk	hallucination defense
Label acceptance	majority of 8, $\geq$3 hits	anti-flicker
Depth trust	$\leq$4.5 m, $\geq$12 samples	LiDAR envelope
Awareness band	3.5–12 m, once per approach	long-range naming
Integrated shadow layer		
Horizontal / vertical bin	10 cm / 10 cm	body-relative vertical evidence
Vertical range	-0.4 to 2.6 m	ground through overhead hazards
Low-confidence hit weight	half weight, capped below confirmed	evidence hierarchy
Traveler defaults	1.75 m height; 0.35 m radius	configurable starting profile
CAPE input / envelope age	0.75 s / 0.5 s	stale evidence cannot back a moving action
Topological / CAPE budgets	2 ms / 5 ms	simulator-instrumented; hardware unverified

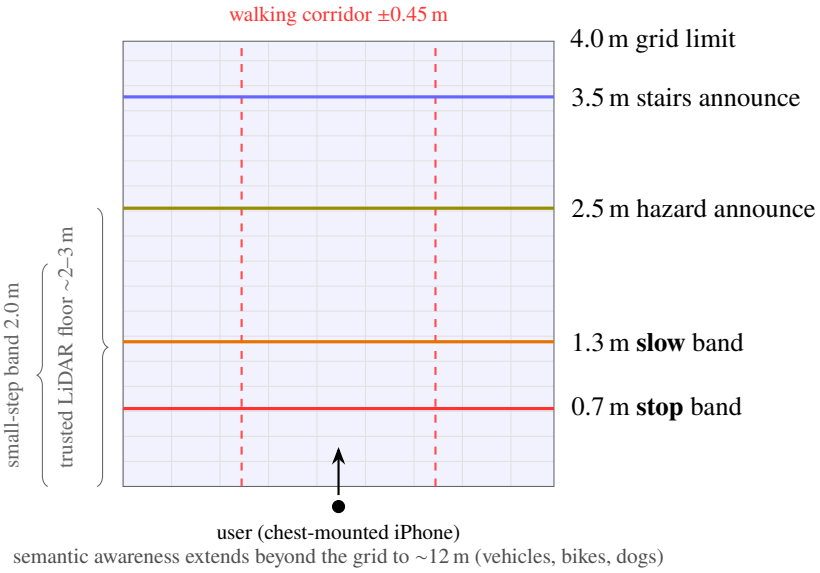


Figure 8: Top-down sensing geometry of the terrain grid (59×27 cells of 7.5 cm) with the calibrated distance bands. Bands are hysteretic (e.g., stop enters at 0.7 m, exits at 0.95 m) to prevent command flip-flopping at boundaries.

as unresolved rather than being presented as fixes.

Table 4: Documented failures, root causes, and resolutions.

Observed failure	**Root cause**	**Resolution**
Floor triggered constant "stop" (Gen. 1)	Region pixel ratios count ground as obstacle	Gravity-aligned unprojection; then full terrain grid (Gen. 3)
Mirrored steering: user told "move left" with clear space right †	Basis vector built as $up \times forward$ (points left); inherited by every lateral consumer	One-line cross-product fix; convention pinned by tests in four subsystems
App froze 0.5 s after launch	Synchronous processing on ARKit delegate queue retained 11 ARFrames, exhausting ARKit's frame pool	Constant-time intake queue; at most one frame in flight; excess dropped
UI froze on target device	≈11 MB/frame CPU image conversion + GPU readback on main thread	GPU-direct camera rendering (RealityKit); conversion path deleted
Overflow crash in drop promotion	Sentinel `Int.min` in row-distance arithmetic; unmasked only after the freeze was fixed	Finite sentinel; lesson: bugs hide behind earlier bugs

Table 4 continued

Observed failure	Root cause	Resolution
"Drop ahead" fired almost constantly †	LiDAR floor returns die at 2–3 m; coverage boundary reads as cliff	Promotion capped by *measured* floor-coverage EMA, corridor-gated
Ground coverage died at 2–3 m †	Sampler discarded all low-confidence depth; grazing-range floor is marked low-confidence	Low-confidence samples reinforce floor only (±12 cm); can never mint hazards
Shallow potholes announced as urgent drops †	12 cm depression met the 12 cm drop threshold	Threshold 15 cm; run-length demotion; drops must span the corridor
Phantom drops on dark/wet pavement †	LiDAR absorption blanks mimic missing floor	Camera-luma veto vs. confirmed-floor patches; fails toward warning
Slopes/crowned roads chattered as steps †	Single-plane floor model; residuals grow with distance	Local quadratic ground surface (Eq. 7), grade-clamped
Small steps (4–12 cm) undetectable †	Cross-slope ±8 cm at 4 m swamps absolute heights	Local row-edge detection, MAD-adaptive threshold, 3-column coherence
Phantom mid-air cells †	LiDAR "flying pixels" interpolated across depth discontinuities	Neighbor-ratio rejection at the sampler
Staircases read as walls; app steered users away from stairs †	Step class covers 10–30 cm; stairs 2+ classify as obstacle; stair rule was dead code	Structural detection: plateau profiles, coherent rises/treads across columns
Descending stairs read as urgent drop †	Missing depth past the top step matches drop signature	Stair columns exempted from drop promotion; descending guidance added
Every object announced "step up ahead" †	10–30 cm band classifies the *base* of any object; step outranked obstacle	Plateau resolution: a step-up must have a walkable surface behind the edge
Sidestepped into a crossing pedestrian's path †	People treated as static terrain	Person tracking with motion classes; never steer in front of a crosser

Table 4 continued

Observed failure	Root cause	Resolution
Receding walker became phantom obstacle trail †	No receding class; person cells entered hazard memory	Leader-following rule; person cells retyped before map ingest
Wall-avoidance turns amplified drift	Half-field openness counted unobserved cells as open	Course-first turns; corridor probes trust only positively observed floor
Magnetometer heading unusable	Chest-mount magnets	GPS-course / AR-displacement offset fusion †; beacon gated on confidence
Detector hallucinated at night †	COCO model on sensor-noise frames	Luma-tiered stand-down; dusk band requires LiDAR corroboration
Core ML export crash at 640×480	Converter failure on attention block at rectangular input (two toolchains verified)	Square 640 + scale-to-fill; distortion cost absorbed by label voting; toolchain pinned
Startup trap with barrier at final grid rows †	Invalid Swift range when obstacle segmentation reached the grid boundary	Boundary-safe iteration plus physical-device regression geometry
Off-grid route candidate trapped during body sweep †	A clamped cell window could still form an inverted closed range when wholly outside the grid	Empty half-open ranges now reach the existing fail-unknown path; straddling and off-grid cases pinned
PRW route candidate silently never generated †	Nested optional flattening collapsed "present and admissible" into the same nil as "absent"	Explicit admissibility state plus a positive route-weave candidate regression
Calibration tool and app rejected each other's artifacts †	Python double and Swift Float32 serialized different quantiles, so hashes could never match	Canonical float32 quantization and cross-language hash fixture
Attention budget could silence an emergency †	Budget rejection ran before the emergency short-circuit	Safety objections remain first, then emergency/health cues bypass attention economics
Stored consent always decoded as corrupt †	Encoder wrote ISO-8601 dates while decoder expected its default format	Symmetric codec and round-trip/revocation tests

Table 4 continued

Observed failure	Root cause	Resolution
Hysteresis changed with thermal load	Promotion and retention were expressed in frame counts while frame rate varied	Convert safety timing to elapsed-time windows; cross-rate verification remains required
Near minor obstacle could hide a farther severe drop	Nearest-first early return ended command evaluation before severity arbitration	Open audit finding: evaluate imminent severity before convenience or route commands
Vehicle-facing green light could be heard as permission to cross	Signal color lacked pedestrian-signal and intersection context	Feature remains disabled; vehicle green must never authorize road entry
Recovery display and guidance health could diverge	Recovery dwell cleared some policy fields before the visible state fully recovered	Re-audit before participant use; UI and command policy must consume one health state

8 Verification and Research Instrumentation

8.1 Automated verification

The pipeline is deliberately structured so that every safety-relevant computation can be exercised without a participant. The current source inventory contains 1,519 `@Test` declarations across 134 Swift test files. In the verification run used for this revision, 1,518 tests passed, one Supabase integration test was skipped because its external test configuration was absent, and none failed. The same run passed secret, experimental-default, guidance-isolation, forbidden-phrase, unsigned-build, and bundle-content audits. Compiler warnings remain, including concurrency isolation warnings, so this is not reported as a warning-free or Swift-6-clean build.

Coverage includes coordinate transforms, terrain and drop gating, people and semantics, local planning, system health, GuideSpace, CAPE, dynamic tubes, topological constraints, route graphs and route belief, maneuver witnesses, counterfactual scans, guided response, attention policy, consented memory, replay, reports, and uploads. Two classes deserve note. First, *invariant tests* exercise rules such as camera semantics never clearing LiDAR occupancy, route confidence never reviving a topological rejection, and shadow types never reaching the guidance bus. Second, *pinned-failure tests* reproduce triggering geometry and logic from diagnosed failures, such as grid-edge range traps, mirrored steering, cross-language calibration hashing, and emergency cues

rejected by attention accounting.

Automated evidence is reported as E2, not as field safety. A test can show that code obeys a specified invariant for its inputs; it cannot show that the specification covers every environment or that a traveler understands the resulting cue. Figure 9 groups the source declarations by test filename to show the distribution of engineering attention; it is an inventory rather than a pass-rate or coverage metric. Table 5 makes the evidence boundary explicit.

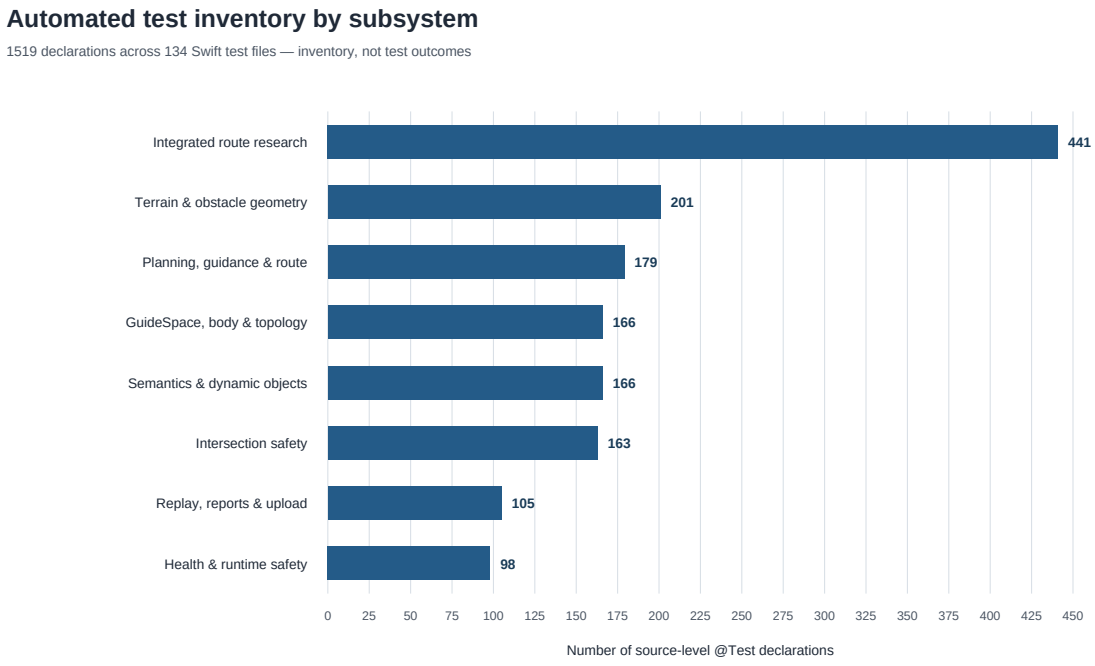


Figure 9: Source-level automated test inventory grouped reproducibly by subsystem filename. Counts describe `@Test` declarations in the repository; they do not report execution results, code coverage, or field safety.

8.2 Field replay: turning walks into experiments

The legacy clip format captures timestamp, camera intrinsics and transform, and compressed depth and confidence maps for each processed frame. ClipV2 extends the schema with pose, motion, GPS, route state, guidance, health, events, compressed RGB, rendered-cue response episodes, and `perceptual_nav.jsonl`. The reader decodes RGB, validates contract versions, and joins streams by frame identity; a record that cannot be joined is counted and refused rather than attached to a neighboring frame. Ordinary clip integrity and subsystem-scoped evidence grade are deliberately separate, so an older clip can still support the measurements it actually contains without becoming evidence for a newer subsystem.

Table 5: Evidence layers used in this paper.

Grade	Evidence	Permitted interpretation
E0	Proposal	"We propose"; no implementation claim
E1	Code or design review	Architecture is designed to behave this way
E2	Unit, synthetic, build, or replay verification	The tested implementation obeyed the stated cases
E3	Developer physical-device observation	The behavior was observed during engineering use, not a controlled participant study
E4	Supervised participant study	Participant outcome claim within the tested operating domain
E5	Independent or longitudinal validation	Generalization beyond the original study team and routes

A *shadow replay* harness runs baseline and candidate configurations over the same recorded input and reports command differences, refusals, and subsystem metrics. Metrics with absent labels return "not measurable," not zero. This is a powerful paired engineering test, but disagreement does not identify which configuration is correct. Human labels, measured course geometry, or observed outcomes remain necessary. Live evaluation also remains necessary because a changed command alters the traveler's next action and therefore the next sensor observation.

8.3 Trial instrumentation

To make the hypothesis measurable (criterion G6), trial mode records monotonic command transitions, route and health state, position and speed, and researcher-marked events. The event vocabulary includes collision, near miss, cane contact, guide or researcher intervention, unexpected stop, ignored or repeated guidance, reversal, wrong turn, overshoot, and successful avoidance. Independent adjudication is still required; the app log supports rather than replaces an observer.

A consent-gated diagnostic ring retains approximately 15 seconds of recent evidence with a bounded memory budget. A one-tap report or conservative automatic trigger can freeze that evidence into a size-limited archive for later analysis. The current research transport replaces the disabled CloudKit path with a provider-neutral manager backed by a private Supabase Storage bucket and metadata table. Anonymous authentication, per-user object paths, and row-level policies are intended to constrain access; the persistent queue retains local work across network loss and falls back to local-only saving when cloud configuration is unavailable. Deployment access control has not yet been independently verified, so this is research infrastructure rather than a safety claim. Live navigation remains offline. RGB and GPS may identify a participant, location, or bystander, so the study protocol must specify consent, access, retention, withdrawal, and redaction.

Figure 10 shows the intended evidence lifecycle. Offline teacher labels are provisional, not ground truth; incidents are useful for active learning only after human adjudication and route/session-separated evaluation.

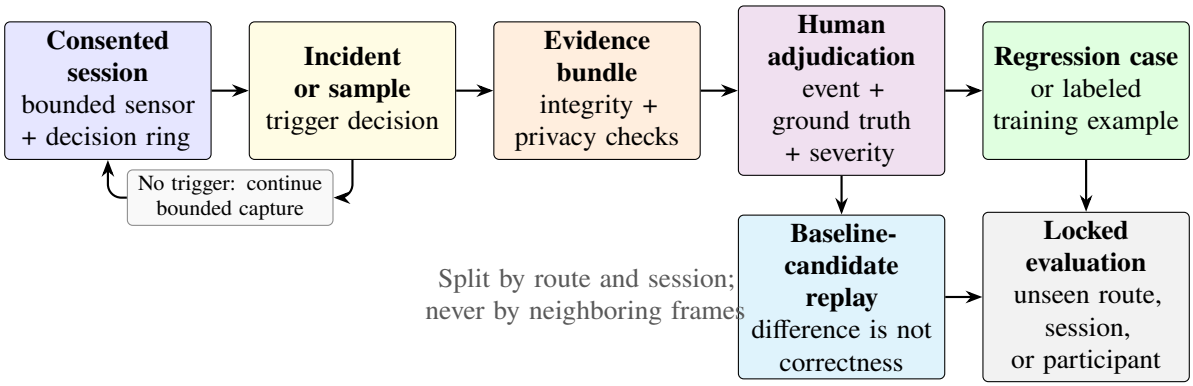


Figure 10: Incident-to-evidence workflow. The locked evaluation set remains separate from training, calibration, and threshold selection.

9 Results

Results are reported against the design criteria of Section 3.3. We report engineering measurements and verified behaviors; controlled human-participant outcomes are explicitly *not* claimed (trials are designed but not yet conducted; Section 10).

G1 — Real time. The single-frame-in-flight architecture removes the capture-pool retention mechanism that caused physical-device freezes, and health instrumentation exposes processing age, dropped work, and recovery state. Developer walks show that thermal and visualization load remain practical constraints. A controlled latency, energy, and thermal benchmark across devices and walking conditions has not yet been completed; therefore no universal frame-rate or p95-latency result is claimed.

G2 — Trustworthy alerts. Coverage-boundary gating, low-confidence floor reinforcement, pothole demotion, photometric continuity checks, and a local ground model address five diagnosed sources of false drop alerts. Unit and synthetic tests establish the intended gates; developer observations motivated the fixes. They do not prove a zero false-alert rate. The stronger current rule is that unknown remains unknown: the system should slow, stop, or request a scan when risk cannot be resolved.

G3 — Terrain completeness. Test-backed logic distinguishes floor, bumps, small and large steps, obstacles, stairs, drops, raised surfaces, and people, including targeted look-alike cases. The shadow stack adds local-ground-relative vertical body bands, uncertainty-expanded dynamic tubes,

and curb-latched pedestrian admissibility. Synthetic scenarios show that a road-side detour can be rejected without reviving another physically rejected path. Controlled detection precision, recall, range error, and warning-distance measurements across lighting, surface, and weather strata remain pending.

G4 — Navigation. GPS-course and AR-displacement fusion, body-relative route cues, an audio beacon, environment scans, A* local planning, and single-voice arbitration are implemented. RouteLock is a default-off map-matching baseline; the route graph, PRW, maneuver machine, counterfactual scan planner, and next-frame route-intent bridge run or refuse inside the isolated shadow stack. G-CAPE and attention selection are implemented but are not yet driven by the live frame path, and the maneuver machine lacks live witness producers. The crossing subsystem is implemented only for shadow recording and cannot reach output; missing validated perception models keep its entry gate closed. Construction no-go polygons and rerouting remain unimplemented.

G5 — Cost and privacy. Live perception and guidance run on a supported iPhone without a network connection, using a commodity mount and audio output. Optional research reports are consent-gated and separated from operation. This is a low-incremental-hardware design, not a zero-cost claim, and privacy depends on a reviewed retention and access policy for RGB and GPS.

G6 — Measurability. Trial events, bounded incident capture, replay, configuration freezing, and shadow comparison are implemented. ClipV2 now decodes RGB and joins typed streams by frame identity. Guidance response episodes, subsystem-scoped evidence gates, refusal-aware metrics, and an adversarial matrix improve traceability, but replay disagreement is not ground truth. External observers, reviewed labels, and blinded event adjudication remain part of the planned participant protocol.

The most transferable engineering findings are: missing sensor data must remain explicit rather than becoming a fabricated drop or a fabricated clear path; local terrain models outperform one global plane on real pavement; camera semantics may annotate but may not clear LiDAR occupancy; moving people require a temporal policy; and every research claim should state whether it is proposed (E0), code-reviewed (E1), test-backed (E2), physically observed by developers (E3), or participant-validated (E4+).

10 Planned Human-Participant Evaluation

The engineering phase makes the hypothesis testable; the next phase tests it. The protocol, modeled on the collision-warning RCT of Pundlik et al. [7] and using the built-in instrumentation of Section 8.3:

- **Staged safety gate.** First complete deterministic adversarial tests and a measured obstacle-course benchmark. Next, trained staff conduct ordinary, non-blindfolded engineering walks with a spotter. Blind and low-vision participant work begins only after SRC/IRB approval, accessible consent, predefined stopping rules, and orientation-and-mobility supervision.

- **Design.** Within-subjects and counterbalanced: each participant retains their usual mobility aid under (a) usual aid alone and (b) usual aid plus PathSense. Audio-only, haptic-only, and combined conditions may be added after the shoe hardware is independently verified. A separate training route precedes scored routes.

- **Operating domain and courses.** Standardized indoor and outdoor routes contain measured obstacles at several heights, curbs, small steps, stairs, controlled drop simulations, and scripted pedestrian interactions. Device, mount, illumination, surface, route, thermal state, and walking speed are logged. Unsupervised street-crossing permission is outside the study domain.

- **Primary outcomes.** Environmental contacts and adjudicated near misses per 100 m; wrong turns and overshoots; experimenter interventions; successful completion. Near-miss criteria are fixed before collection and reviewed independently. Completion time, preferred-speed fraction, false stops, command reversals, and an accessible workload measure are secondary or guardrail outcomes.

- **Analysis.** Pre-register the primary outcome and use a pilot to estimate variance and power. Count outcomes can be analyzed with a mixed-effects Poisson or negative-binomial model including route exposure and participant effects; continuous outcomes use paired or mixed-effects models after distribution checks. Report effect sizes and confidence intervals, not only significance tests.

- **Configuration and data integrity.** Freeze the complete feature configuration per session. Split model-development data by participant, route, and session—never neighboring frames. Keep training, calibration, locked validation, and final test sets separate. Shadow replay supports ablation but cannot replace live evaluation.

- **Stop rules.** Pause if the spotter intervenes, processing or mount health becomes stale, localization conflicts with the physical corridor, repeated contradictory commands occur, the participant requests a stop, or a crossing would require the system to grant permission.

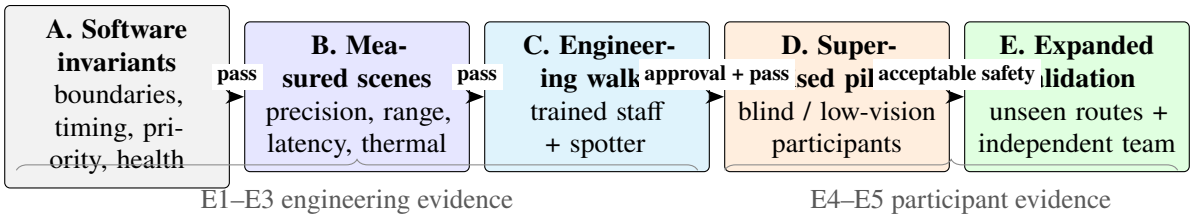


Figure 11: Evaluation proceeds only after each preceding safety gate is passed. Engineering tests and developer walks cannot be reported as participant outcomes.

11 Discussion

11.1 What the failures teach

Four findings generalize beyond this project. First, **missing sensor data require their own state**. Treating no LiDAR return as a drop created phantom cliffs; treating it as free space would be more dangerous. Unknown should remain unknown until positive evidence or a conservative slow/stop/scan policy resolves it. Second, **authority must be architectural, not a confidence threshold**. Camera semantics can name objects and add conservative cost, but they cannot clear LiDAR occupancy. Third, **local models outperform one global model on real pavement**. A single plane failed on crowns and slopes, while absolute step heights failed against cross-slope; local surfaces and local height discontinuities matched the actual geometry. Fourth, **safety timing must be expressed in elapsed time**. A frame-count dwell changes behavior under thermal throttling and dropped frames, so promotion, retention, and recovery windows require cross-rate verification.

11.2 Relation to prior work

Pundlik et al. established that wearable collision warning reduces contacts [7]; PathSense extends the concept from binary warning toward terrain classification, wayfinding, and semantic awareness on a mainstream phone—directly addressing cost and adoption constraints cataloged by the surveys [3, 6, 4]. The feedback design operationalizes the haptics literature [11]: pattern-coded, language-independent, low-attention channels for continuous guidance, with speech reserved for symbolic content. The cognitive-overload warning of Real and Araujo [3] shaped the strictest user-facing rule in the system: one voice, one command, once per approach.

NaviSense demonstrates a contemporary multimodal combination of mobile vision, AR, LiDAR, and audio–haptic feedback with blind and low-vision participants [13]; a recent smartphone navigation preprint likewise reports participant outcomes using scene understanding [21]. These systems make a first-of-kind claim inappropriate. PathSense instead emphasizes terrain-specific failure modes,

source-authority rules, body-height evidence, and a reproducible incident-to-regression workflow. CAPE is also positioned as an extension to, not a replacement for, conformal navigation research [14, 15].

11.3 Limitations

The central limitation is that **no controlled blind or low-vision participant outcome data yet exist**; all current safety statements are engineering claims, not clinical or mobility-efficacy claims. Much of the stack has E2 synthetic or unit evidence, selected failures have E3 developer observations, and a large field-verification checklist remains open.

Sensor physics bounds the geometry: grazing-angle LiDAR coverage is short and nonuniform; transparent, reflective, dark, wet, rain-covered, and snowy surfaces remain difficult. The detector is COCO-bound, monocular class-height priors vary by instance, and the square model input introduces aspect distortion. Chest gait, thermal load, pose drift, GPS/heading error, and moving-agent intent remain sources of uncertainty. Development centered on one phone model and one mount family.

The active planner still uses the legacy terrain grid. GuideSpace, body slices, CAPE, dynamic tubes, the topological field, route graph, PRW, and counterfactual scanning are default-off shadow systems. G-CAPE has no real calibration artifact and is not driven from the live frame path; the maneuver machine lacks live witnesses; the attention policy is not connected to live cue alternatives; and route memory has no field evidence. The crossing subsystem is shadow-only and lacks validated pedestrian-signal, crosswalk, curb-ramp, and roadway models. Construction closures and rerouting remain unimplemented. Gait locking and temporal evidential learning remain E0 ideas. Replay is identity-aligned and RGB-capable, but baseline–candidate agreement still cannot establish correctness. Finally, pulse audio occupies a channel blind travelers rely on; the proposed haptic shoes have not yet been evaluated.

11.4 Future work

Immediate work is to (1) collect reviewed ground truth and controlled-scene benchmarks; (2) complete the approved, supervised evaluation sequence in Figure 11; (3) produce route-, session-, environment-, and participant-separated calibration data for G-CAPE; (4) add live, validated maneuver witnesses and wire G-CAPE and attention decisions into recording without granting output authority; (5) implement construction closures, no-go polygons, and safe rerouting; (6) validate pedestrian-signal, crosswalk, curb-ramp, and roadway models before any crossing output; (7) measure latency, energy, and thermal behavior on supported hardware; and (8) prototype and

independently evaluate the haptic shoes. Gait-aware sampling and temporal evidential learning remain separate candidate experiments after the measurement foundation is stable.

12 Conclusion

PathSense establishes a testable, smartphone-based architecture for terrain geometry, semantic context, route intent, governed feedback, and research replay. Its development record supports five engineering lessons: preserve unknown evidence; fit terrain locally; keep camera semantics from clearing LiDAR occupancy; model moving people temporally; and separate test-backed, field-observed, and participant-validated claims. GuideSpace extends the representation to vertical, traveler-relative risk; the default-off research stack adds body-swept candidates, dynamic tubes, topological curb memory, multi-hypothesis route belief, cue-conditioned response envelopes, maneuver witnesses, decision-relevant scans, attention budgeting, and consented local memory. These implementations are reported as E2 shadow software with explicit refusals, not as active safety features or validated inventions. The hypothesis that PathSense can improve mobility outcomes when added to a traveler's usual aid remains open and will be answered only through the staged participant protocol.

Acknowledgments

We thank our adult sponsor, Ethan Dusto, and Cherry Creek High School for supporting this project. Generative AI coding assistants, including Anthropic Claude Code and OpenAI Codex, were used for implementation suggestions, refactoring, test generation, debugging, design critique, documentation, and research-record synthesis. The student researchers defined the research questions and requirements, reviewed generated material, ran and interpreted tests, and remain responsible for the code, citations, safety decisions, data, and manuscript. No external funding was received. No human participants were involved in the work reported here; planned participant trials will proceed only after SRC/IRB review.

References

[1] World Health Organization, "World report on vision," tech. rep., World Health Organization, Geneva, Switzerland, 2019. Licence: CC BY-NC-SA 3.0 IGO.

[2] P. Ackland, S. Resnikoff, and R. Bourne, "World blindness and visual impairment: despite many successes, the problem is growing," *Community Eye Health*, vol. 30, no. 100, pp. 71–73, 2017.

[3] S. Real and A. Araujo, "Navigation systems for the blind and visually impaired: Past work, challenges, and open problems," *Sensors*, vol. 19, no. 15, p. 3404, 2019.

[4] F. E.-Z. El-Taher, L. Miralles-Pechuán, J. Courtney, K. Millar, C. Smith, and S. Mckeever, "A survey on outdoor navigation applications for people with visual impairments," *IEEE Access*, vol. 11, pp. 14647–14666, 2023.

[5] R. Manduchi and S. Kurniawan, "Mobility-related accidents experienced by people with visual impairment," *AER Journal: Research and Practice in Visual Impairment and Blindness*, vol. 4, no. 2, pp. 44–54, 2011.

[6] M. D. Messaoudi, B.-A. J. Menelas, and H. Mcheick, "Review of navigation assistive tools and technologies for the visually impaired," *Sensors*, vol. 22, no. 20, p. 7888, 2022.

[7] S. Pundlik, V. Baliutaviciute, M. Moharrer, A. R. Bowers, and G. Luo, "Home-use evaluation of a wearable collision warning device for individuals with severe vision impairments: A randomized clinical trial," *JAMA Ophthalmology*, vol. 139, no. 9, pp. 998–1005, 2021.

[8] Apple Inc., "ARKit developer documentation: ARFrame, sceneDepth, and ARPlaneAnchor." `https://developer.apple.com/documentation/arkit`, 2024.

[9] Apple Inc., "Core ML and the Apple Neural Engine: Deploying machine learning models on-device." `https://developer.apple.com/documentation/coreml`, 2024.

[10] G. Jocher and J. Qiu, "Ultralytics YOLO11." `https://github.com/ultralytics/ultralytics`, 2024. Version 11.0.0, AGPL-3.0 license.

[11] S. Khusro, B. Shah, I. Khan, and S. Rahman, "Haptic feedback to assist blind people in indoor environment using vibration patterns," *Sensors*, vol. 22, no. 1, p. 361, 2022.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014*, pp. 740–755, Springer, 2014.

[13] A. N. Sridhar, F. Qiao, N. D. Troncoso Aldas, Y. Shi, M. Mahdavi, L. Itti, and V. Narayanan, "NaviSense: A multimodal assistive mobile application for object retrieval by persons with visual impairment," in *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility*, Association for Computing Machinery, 2025.

[14] A. Dixit, Z. Mei, M. Booker, M. Storey-Matsutani, A. Z. Ren, and A. Majumdar, "Perceive with confidence: Statistical safety assurances for navigation with learning-based perception." arXiv:2403.08185, 2024.

[15] J. Shin, J. Lee, and I. Yang, "Egocentric conformal prediction for safe and efficient navigation in dynamic cluttered environments." arXiv:2504.00447, 2025.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[17] K. V. Mardia and P. E. Jupp, *Directional Statistics*. Chichester, UK: John Wiley & Sons, 2000.

[18] Federal Highway Administration, *Manual on Uniform Traffic Control Devices for Streets and Highways, 11th Edition*. U.S. Department of Transportation, 2023.

[19] U.S. Access Board, *Accessibility Guidelines for Pedestrian Facilities in the Public Right-of-Way*. Architectural and Transportation Barriers Compliance Board, 2023.

[20] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[21] R. Liu and P. Slade, "Improving outdoor navigation for people with blindness using an AI-driven smartphone application and personalized audio guidance." arXiv:2605.29120, 2026.