Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Wearable Non-Visual Guidance System with an Inspectable Spatial-Cue Core

Chenghang Li
Yirui Yuan
Shurui Zhang
Wenxuan Ma
17888564210@163.com

Fan Wang
Yang Gao

Research Article

Keywords:

Posted Date: November 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11251351/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Wearable Non-Visual Guidance System with an Inspectable Spatial-Cue Core

Chenghang Li[1,†], Yirui Yuan[1,†], Shurui Zhang[1,†]
Wenxuan Ma[2,*], Fan Wang[3,*], Yang Gao[1,*]
[1]South China University of Technology, Guangzhou, China
[2]School of Biomedical Engineering, Southern Medical University
[3]School of Information Management, Wuhan University
Corresponding authors: Wenxuan Ma, 17888564210@163.com;
Fan Wang, 2021101040028@whu.edu.cn;
Yang Gao, gaoyang2025@scut.edu.cn
Author emails: Chenghang Li, koji3527@qq.com; Yirui Yuan, 15312077517@163.com;
Shurui Zhang, 3251573076@qq.com; Yang Gao, gaoyang2025@scut.edu.cn
Equal contribution: Chenghang Li, Yirui Yuan, and Shurui Zhang

Abstract

Wearable assistance for blind and low-vision users must convert egocentric perception into interpretable action cues. This paper presents a three-node guidance system comprising smart glasses, an RK3588 edge unit, and a four-motor wristband. RGB-D/stereo perception, YOLO11s detection, metric depth estimation, and 3D back-projection produce task-specific spatial relations. An upstream cue core validates task intent, entity, mode, relation, and state before transmitting vector packets over BLE; the wristband applies roll compensation and renders directional haptic feedback. Placement-level relative-vector errors are 1.13 ± 0.38, 1.66 ± 0.47, and 1.76 ± 0.30 cm at nominal 30, 50, and 70 cm placements ($n = 15$ per distance). Grasping succeeds in 37/40 trials under a 60 s timeout. A supplementary study report compares smart glasses and a white cane in 100 participants, including 50 blind and 50 low-vision participants, using a counterbalanced mixed design and collision counts as the primary outcome. In the blind group, mean collision counts are 4.46 with glasses and 12.54 with the cane; the report gives $t(49) = 30.92$, $p < 0.001$, and Cohen's $d = 4.37$. In the low-vision group, the mean paired difference is 0.14 collisions, with a reported 90% confidence interval of $[-0.01, 0.29]$, meeting the specified ± 1.5-collision equivalence criterion. The reported age-adjusted group-by-device interaction is significant ($F(1, 97) = 412.5$, $p < 0.001$), supporting visual-status-dependent benefit. By integrating performance indicators and behavioral outcomes, this multidimensional framework provides a practical tool for evaluating wearable assistive systems, supports safer and more effective autonomous assistance, and contributes to broader human–machine interaction research.

Keywords: wearable assistive technology; RGB-D perception; metric depth estimation; spatial cue generation; haptic feedback; human–machine interaction

1 Introduction

Blind and low-vision (BLV) assistance requires more than recognizing objects: visual evidence must be converted into guidance that is timely, locally referenced, and understandable without vision.

The long white cane remains a robust and inexpensive mobility aid because it provides direct tactile contact with nearby obstacles, but it does not provide remote object identity, depth-supported reachability, or task-specific directional guidance. Electronic travel aids extend the sensing range with cameras, depth sensors, audio, and haptic displays. Wearable travel-aid systems have addressed environment perception and navigation [2]; product-localization systems have supported finding requested items [18]; and automated hand-navigation systems have explored tactile assistance during grasping [10]. A systematic review further shows how haptic feedback can encode spatial information without continuously occupying the auditory channel [13]. These efforts establish the practical value of sensing and non-visual feedback, but they also expose the key translation problem addressed in this paper: a detected object or obstacle is not yet an executable local cue.

Several research directions have advanced the perception side of this problem. Egocentric datasets such as Ego4D and EPIC-KITCHENS have supported recognition of objects, actions, and activities from the user's viewpoint [11, 6, 7]. More recent smart-glasses agents and multimodal assistive systems have added speech interaction, open-ended visual question answering, product localization, and object-retrieval assistance [16, 22, 18]. Their strength is broad semantic understanding: the system can identify what is present and respond to a user's request. For grasping or obstacle negotiation, however, labels and natural-language descriptions do not fully specify where the user's hand or body should move, whether the estimate is expressed in a stable coordinate frame, or whether the underlying depth evidence is still valid.

Other studies have focused on the feedback side. Tactile bracelets and haptic navigation systems have demonstrated that directional and urgency information can be delivered through compact wearable interfaces [10, 21, 13]. Edge-AI research further motivates local processing when feedback depends on predictable latency and network-independent operation [19]. These approaches provide important building blocks, but a system may still fail at the interface between them. A raw vector can be incomplete, stale, or associated with the wrong task mode; a camera-frame direction can become misleading when the wrist rotates; and a cloud fallback can make the timing of a safety-relevant cue difficult to control. What is often missing is an explicit, testable middle state that records the task, grounded entity, spatial relation, validity, and feedback action before actuation.

The objective of this work is therefore to design and verify that middle state as a system component. We implement a three-node wearable prototype consisting of smart glasses, an RK3588 edge computer, and a four-motor wristband. The edge node combines task grounding, object or hazard perception, and depth-supported geometry. The spatial-cue core then converts these inputs into a structured relation, checks entity, mode, completeness, and freshness, compensates wrist roll, and emits a compact audio-haptic command only when the cue is valid. The same interface supports hand-to-object guidance in grasp mode and local lateral-forward hazard guidance in avoidance mode. The overall workflow is shown in Fig. 1; the middle representation denoted there as actionable spatial intention is the spatial-cue record formalized in Section 4.

The contributions of this work are:

- A three-node wearable BLV guidance architecture with a dedicated spatial-cue core between egocentric perception and multimodal feedback.
- A formal and implementable vision-to-cue pipeline that back-projects depth-supported detections into metric 3D vectors, compensates wrist posture, and suppresses invalid or stale cues.
- A system-level verification framework covering perception and geometry, cue-state fault replay, haptic rendering, integrated grasping, and local obstacle avoidance.

The remainder of the paper describes the system architecture, the spatial-cue formulation and embedded implementation, and the verification results and limitations of the prototype.

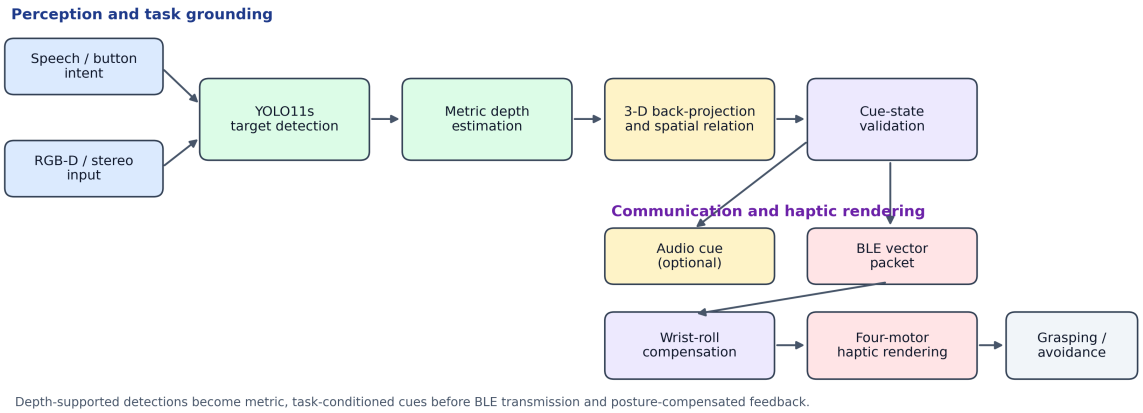


Figure 1: Overall workflow of the proposed wearable assistance system. Multimodal context and task grounding are converted into the spatial-cue record, which is rendered as audio-haptic feedback and updated through user action.

Table 1: Comparison of assistive approaches considered in the system design.

Approach	Sensing range	Intermediate state	Feedback	Main limitation
White cane	Contact range	None	Direct tactile contact	No remote perception or object identity
Guide dog	Surrounding environment	Handler–dog interaction	Motion guidance	High cost and limited logging
Speech-only wearable	Camera range	Label or caption	Audio	Weak continuous direction cue
Direct haptic mapping	Camera/depth range	Raw vector	Vibration	Hard to distinguish invalid and complete states
Proposed system	Camera, depth, pose	Spatial-cue contract	Audio plus haptics	Requires larger user validation

2 Related Work

Related systems address three tasks: sensing the environment, identifying the requested target, and communicating guidance. This section compares how they connect these tasks.

2.1 Assistive sensing and object retrieval

Electronic travel aids extend the contact range of conventional mobility tools by combining cameras, depth or range sensing, and user feedback. Bai et al. described a wearable travel aid for environment perception and navigation, while more recent work has investigated product localization and automated hand guidance for object retrieval [2, 18, 10]. These systems demonstrate that distant targets can be detected and that a wearable device can assist reaching. The present work uses the same broad sensing-to-assistance direction, but focuses on the intermediate representation that decides whether a detected relation is ready to be rendered.

2.2 Egocentric perception and task grounding

Large egocentric datasets and multimodal models have improved recognition of objects, activities, and user intent [11, 7, 20]. Smart-glasses agents further combine vision, speech, and cloud services to answer questions or provide contextual assistance [16, 22, 21]. From the perspective of the present system, these methods provide evidence and task semantics, but do not by themselves define a stable metric relation for human movement. Robotics and embodied AI research has similarly emphasized grounding perception in affordances and action feasibility rather than only semantic labels [1, 8, 4]. We adapt this principle to wearable human guidance: the output is not an autonomous robot action, but an inspectable relation that can be logged and rendered as a cue.

2.3 Edge AI and haptic feedback

Edge computing is useful for latency-sensitive assistive devices because it reduces dependence on network connectivity and keeps frequent perception local [19]. Haptic interfaces can encode direction and urgency without continuously occupying the auditory channel [13]. Existing work therefore supplies two complementary design choices: local computation for timing and tactile output for compact feedback. The contribution of this paper is the interface between those choices. The spatial-cue core explicitly binds task mode, grounded entity, metric relation, validity, freshness, and output command, and is evaluated independently from the perception models and motor drivers.

2.4 Positioning of the present system

Table 1 summarizes the positioning. Conventional aids provide robust local contact, perception-oriented wearables provide remote scene evidence, and haptic systems provide a compact output channel. The proposed system is distinguished by making the perception-to-feedback state explicit and by using the same typed relation for both grasping and local obstacle avoidance.

3 System Architecture

3.1 Design requirements

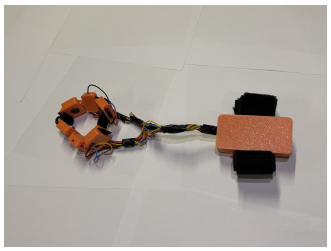
The target use case is short-range non-visual assistance during object retrieval and local obstacle negotiation. The system is designed around five requirements. First, the device should support hands-light interaction so that the user can reach or walk without holding a phone. Second, perception should be processed locally whenever latency and network stability matter. Third, the representation between perception and feedback should be inspectable, because a silent cue may mean task completion, missing depth, stale evidence, or a suppressed hazard. Fourth, haptic output should encode direction without occupying the auditory channel continuously. Fifth, the system should expose module boundaries that can be tested independently.

3.2 Three-node wearable layout

The prototype contains three physical nodes: smart glasses, an edge-computing unit, and a haptic wristband. The glasses node contains the camera, microphone, speaker, and button interface. The edge node receives egocentric frames and interaction events, performs local perception and optional cloud fallback, and emits a compact vector packet. The wristband node receives the packet, estimates roll posture, solves the spatial-to-haptic mapping, and drives four vibration motors.

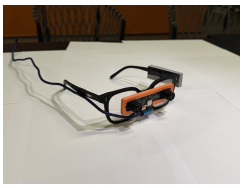


(a) Glasses worn by the operator



(b) Haptic wristband

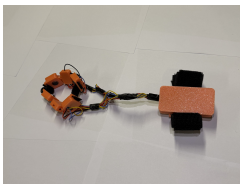
Figure 2: Current wearable prototype used for the reported engineering trials. The photographs show the updated smart-glasses appearance and the four-motor wristband assembly.



(a) glasses node



(c) edge-computing enclosure



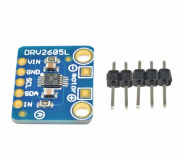
(e) wristband node



(b) stereo camera



(d) RK3588 board



(f) haptic driver module

Figure 3: Real-image hardware composition of the three-node prototype. Representative physical modules are shown together with the communication and control boundaries described in the text.

The communication design intentionally keeps the inter-node message small. The glasses-side controller exchanges framed audio, button, mode, and status messages with the host. The edge-to-wristband interface transmits the action-relevant mode, vector components, distance, activity flag, and status. The wristband assigns a local reception time and sequence number; source-side freshness belongs to the upstream cue-state contract. This keeps the wristband independent of the visual model and lets the cue solver be replayed from logs.

3.3 Fail-silent guidance policy

The spatial-cue contract assigns task grounding, depth, vector completeness, mode, activity, and freshness checks to the upstream cue core. A failed check produces an inactive cue state. The wristband handles packet reception, wrist-roll compensation, and motor-command rendering. This division keeps perception-dependent decisions in the upstream module and tactile execution in the wearable endpoint. The upstream validity rules are evaluated using the cue-state replay in Section 6.

3.4 Runtime modes

Two runtime modes are implemented. In grasp mode, a spoken or selected target object is grounded in the camera frame, paired with the hand position, and converted into a hand-to-target vector. The wristband guides the hand until the target is close enough, after which vibration is muted to

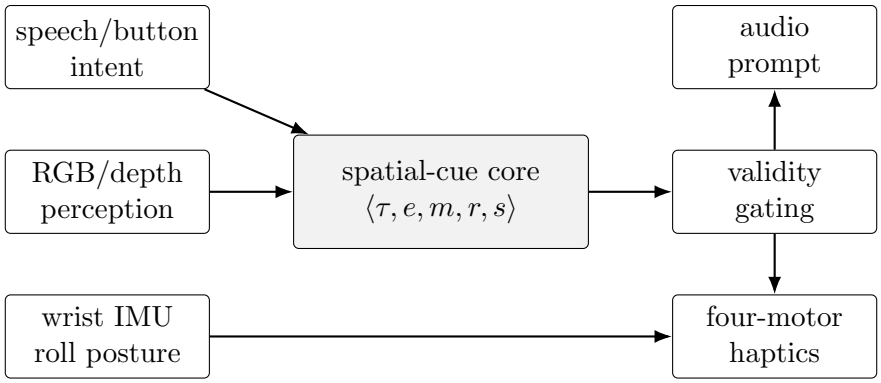


Figure 4: System data flow. The central module is an explicit upstream spatial-cue contract; wrist pose is consumed downstream by the haptic renderer rather than included in the cue-state tuple.

avoid overshooting. In avoid mode, detected hazards are ranked by class priority and distance, then converted into a lateral-forward warning vector. The same packet structure is used in both modes, but the geometric meaning of the relation field is mode-specific.

4 Spatial-Cue Core and Visual Geometry

4.1 Cue-state formulation

At time t, the upstream perception module receives an egocentric observation x_t, a user/task state u_t, and recent interaction history h_t. It produces an inspectable cue state rather than a recognition-only output:

$$C_t = F(u_t, x_t, h_t). \tag{1}$$

The upstream cue state is represented as

$$C_t = \langle \tau_t, e_t, m_t, \mathbf{r}_t, s_t \rangle, \tag{2}$$

where τ_t is the task intent, e_t is the grounded entity, m_t is the execution mode, $\mathbf{r}_t$ is the metric spatial relation, and s_t is the validity state. The spatial-cue core is therefore the middle module that connects perception to feedback. The wristband is a downstream renderer: given a valid C_t and wrist-roll estimate ϕ_t, it computes $\mathbf{a}_t^h = H(C_t, \phi_t)$; speech is rendered from the intent and entity fields. Wrist pose and motor amplitudes are not fields in the upstream cue state.

4.2 Vision-to-3D-vector pipeline

The edge node converts image-space perception into a metric vector before sending it to the wristband. Let the calibrated camera matrix be

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

For any valid pixel (u, v) with metric depth $D(u, v)$, the corresponding 3D camera-frame point is obtained by pinhole back-projection:

$$\mathbf{p}^c(u, v) = D(u, v)\mathbf{K}^{-1}[u, v, 1]^\top. \tag{4}$$



(a) smart glasses (b) edge-computing node (c) haptic wristband

Figure 5: The three physical nodes that execute the shared spatial-cue pipeline. The smart glasses acquire egocentric input, the edge node forms a task-specific 3D relation, and the wristband renders the relation as posture-compensated haptic guidance. The device photographs were background-separated for a consistent presentation.

Table 2: Spatial-cue fields and their runtime meanings.

Field	Meaning	Runtime realization	Grasp mode	Avoid mode
τ	User task	spoken intent or selected mode	requested object	local obstacle avoidance
e	Grounded entity	target/hazard class and detection id	object plus hand	nearest valid hazard
m	Mode	`grasp` or `avoid`	hand-centered reaching	local camera/wrist-referenced warning
$\mathbf{r}$	Spatial relation	metric $x/y/z$ vector and distance	target minus hand	obstacle direction and distance
s	Validity	tracking, stop, active flag, freshness	active until near target	active while hazard is valid
$\mathbf{a}^h$	Downstream haptic command	wristband renderer output	directional hand cue	directional warning cue

The camera frame uses x rightward, y downward, and z forward. Depth may come from stereo inference or a monocular/depth fallback model; detection may come from local object detection, segmentation, or a grounded multimodal model depending on the mode [14, 12, 5, 15, 3].

For the bounding-box implementation, let (u_Ω, v_Ω) be the box center and Ω' its central depth-sampling region. A robust scalar depth is back-projected along the center ray:

$$\hat{z}_\Omega = \text{median}_{(u,v)\in\Omega'} D(u,v), \qquad \mathbf{p}^c_\Omega = \hat{z}_\Omega \mathbf{K}^{-1}[u_\Omega, v_\Omega, 1]^\top, \tag{5}$$

where only valid depth samples contribute to the median. If the valid-depth ratio in the sampling region is below its threshold, the relation is marked incomplete. This box-center estimate is not necessarily an anatomical hand point or an object center; the reference-placement experiment uses its separately specified corresponding measurement points.

In grasp mode, the edge node estimates the target center $\mathbf{p}^c_{obj}$ and the hand or palm center $\mathbf{p}^c_{hand}$. The relation sent to the cue core is

$$\mathbf{r}^g_t = \mathbf{p}^c_{obj,t} - \mathbf{p}^c_{hand,t} = [x_g, y_g, z_g]^\top. \tag{6}$$

The distance field is

$$d^g_t = \|\mathbf{r}^g_t\|_2. \tag{7}$$

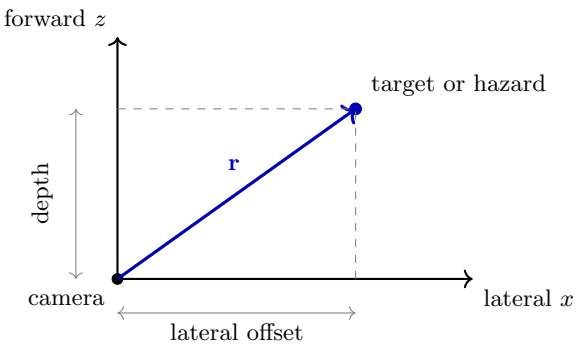


Figure 6: Metric vector construction after depth back-projection. Grasping uses the vector from hand to target, whereas avoidance uses the obstacle vector projected to the lateral-forward plane.

When $d_t^g < \epsilon_g$, the state is set to `stop`; the wristband then suppresses directional vibration so the user is not guided past the object.

In avoid mode, each hazard candidate i has class c_i, confidence γ_i, distance d_i, and center point $\mathbf{p}_i^c$. The candidate score is

$$\rho_i = \frac{\lambda(c_i)\gamma_i}{d_i + \epsilon}, \tag{8}$$

where $\lambda(c_i)$ is a class-priority weight and ϵ prevents division by zero. The selected hazard is

$$i^* = \arg\max_i \rho_i. \tag{9}$$

The avoid relation preserves the lateral and forward components,

$$\mathbf{r}_t^a = [x_{i^*}, 0, z_{i^*}]^\top, \quad d_t^a = \sqrt{x_{i^*}^2 + z_{i^*}^2}. \tag{10}$$

This is a local warning vector, not a global route planner. It tells the user where the immediate hazard lies relative to the body/camera direction.

4.3 Validity gating

The cue core only activates haptic feedback when the entity, relation, mode, and timing are mutually consistent:

$$s_t = \text{active} \iff E_t \wedge R_t \wedge M_t \wedge A_t \wedge (\Delta t < T_{stale}), \tag{11}$$

where E_t denotes successful entity grounding, R_t complete vector components and valid depth, M_t mode consistency, A_t an active vector flag, and T_{stale} the freshness limit. If any condition fails, the system returns an inactive cue state and directional vibration is not emitted. The inactive state is still logged with its cause, which is important for later fault analysis.

4.4 Wrist roll compensation

The wristband is not fixed in a global orientation. A cue that is correct in the camera frame may feel rotated if the arm turns. The IMU therefore estimates wrist roll from accelerometer readings. Using accelerometer components a_y and a_z, the firmware computes

$$\sin\phi_t = \frac{a_y}{\sqrt{a_y^2 + a_z^2}}, \quad \cos\phi_t = \frac{a_z}{\sqrt{a_y^2 + a_z^2}}. \tag{12}$$

Algorithm 1 Vision-to-cue-state generation

Require: RGB frame, depth map, camera matrix, task mode, user intent
Ensure: upstream cue state $C = \langle \tau, e, m, \mathbf{r}, s \rangle$

```
detect task-relevant targets, hands, or hazards
estimate robust 3D positions by depth back-projection
if mode is grasp then
    set r ← p^c_obj − p^c_hand
else
    rank hazards by class priority, confidence, and distance
    set r ← [x_{i*}, 0, z_{i*}]^⊤
end if
check entity, depth, vector completeness, mode, activity, and freshness
if the cue state is invalid then
    set s ← inactive and suppress directional haptics
else
    set s ← active and emit the vector packet
end if
pass valid cue state to the downstream audio/haptic renderers
```

The estimate is exponentially smoothed:

$$\hat{s}_t = (1-\alpha)\hat{s}_{t-1} + \alpha \sin\phi_t, \quad \hat{c}_t = (1-\alpha)\hat{c}_{t-1} + \alpha \cos\phi_t, \tag{13}$$

with $\alpha = 0.18$ in the firmware. For the lateral-forward components, let $\mathbf{r}_{plane,t} = [r_x, r_z]^\top$. A valid relation is rotated into the wrist frame as

$$\tilde{\mathbf{r}}_t = \begin{bmatrix} \hat{c}_t & -\hat{s}_t \\ \hat{s}_t & \hat{c}_t \end{bmatrix} \mathbf{r}_{plane,t}. \tag{14}$$

The avoid branch quantizes roll into eight directions before mapping the vector to motors. The grasp branch uses the smoothed roll components directly in its corner-conjugate mapping. In the current prototype, grasping cannot be completed without posture compensation; compensation was therefore retained as an operational prerequisite in all grasping trials, rather than evaluated as an optional ablation. This observation is specific to the present configuration and does not provide a numerical estimate of a compensation benefit. Roll correction addresses wrist rotation about one axis, not unrestricted camera-to-arm orientation.

4.5 Four-channel haptic synthesis

The wristband has four tactile channels located at right-up, right-down, left-down, and left-up positions. Each valid vector component is clamped to ± 0.20 m and normalized. Directional vibration is generated by distributing the normalized vector across the four corner motors. Distance changes the global intensity:

$$\eta(d) = \begin{cases} 1, & d \le d_n \text{ or } d \ge 100, \\ \eta_{min} + \dfrac{d_f - d}{d_f - d_n}(1 - \eta_{min}), & d_n < d < d_f, \\ \eta_{min}, & d_f \le d < 100, \end{cases} \tag{15}$$

Algorithm 2 Wristband cue solving and playback

Require: vector packet, IMU roll sample
Ensure: motor amplitudes and pulse timing
initialize all motor amplitudes to zero
if packet is missing or inactive **then**
    stop all motors and return
end if
if mode is grasp and $d < 0.05$ m **then**
    stop all motors and return
end if
if mode is avoid and $d < 0.05$ m **then**
    set all motors to danger pattern and return
end if
rotate relation into wrist frame using roll estimate
map planar direction to four corner motor amplitudes
apply distance gain and amplitude clipping
play mode-specific pulse pattern

where $d_n = 0.05$ m, $d_f = 1.00$ m, and $\eta_{min} = 0.25$. The $d \geq 100$ branch describes a legacy distance sentinel in the firmware, not a measured long-range operating condition; upstream validation must establish a meaningful metric relation. The final motor command is

$$\mathbf{a}_t^h = \eta(d_t)\,\mathrm{clip}\{G(\tilde{\mathbf{r}}_t), 0, P_{max}\}, \tag{16}$$

where $P_{max} = 95$ is the real-time playback amplitude limit and $G : \mathbb{R}^2 \to \mathbb{R}^4$ maps the normalized lateral-forward direction to the four corner-motor amplitudes. Directional vectors are rendered as one 140 ms pulse; front, back, and danger states use distinct pulse counts and intervals.

5 System Implementation

5.1 Glasses and interaction controller



(a) glasses assembly **(b)** controller enclosure

Figure 7: Smart-glasses assembly used for sensing and user interaction.

The glasses-side controller is implemented on an ESP32-C3-class microcontroller using ESP-IDF [9]. It manages button events, mode switching, microphone recording, speaker playback, and serial framing. The host protocol defines separate message types for connection status, button events, mode changes, audio upload, audio download, playback control, status query, and diagnostic logs.

Table 3: Edge-to-wristband vector packet used by the spatial-cue core.

Field	Type	Function
`mode`	enum/string	selects grasp or avoid geometry
`status`	enum/string	marks tracking, stop, or inactive state
`vector_active`	Boolean	enables directional rendering only when true
`vector_x_m`	float	lateral relation in metres
`vector_y_m`	float	vertical relation in metres for grasping
`vector_z_m`	float	forward-depth relation in metres
`distance_m`	float	distance gain and stop/danger threshold
Local sequence/tick	receiver metadata	identifies new input and records its reception time

This separation allows the perception host to react to user input while audio is being recorded or replayed.

The user can enter three high-level states: idle/voice interaction, grasping assistance, and navigation/avoidance assistance. Mode changes clear previous task vectors before a new cue state is activated. This prevents a stale target vector from being reused after the user switches from object retrieval to obstacle warning.

5.2 Edge perception and task grounding



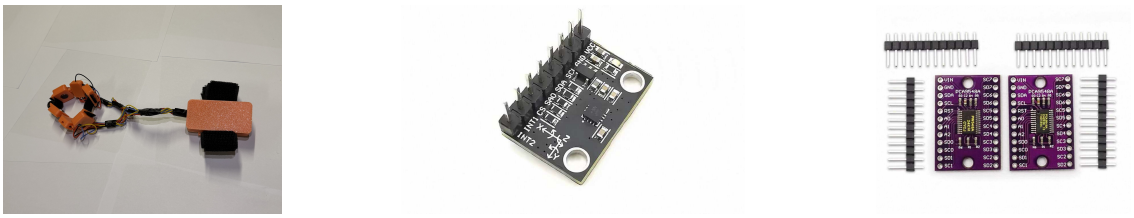
RK3588 edge-computing board

Figure 8: Clean-background photograph of the RK3588 edge-computing board used by the perception and task-grounding pipeline.

The edge node runs local perception on an RK3588-class platform. Object detection provides candidate targets and hazards, while stereo or learned depth estimation provides metric depth maps. Open-category grounding and fallback segmentation can be handled by cloud or larger multimodal models when the requested object is outside the local detector vocabulary. The output of this stage is not a full scene description; it is a compact vector packet matching the spatial-cue fields in Table 2.

The packet contains the mode, status, vector activity flag, $x/y/z$ vector components in metres, and distance in metres. The receiver assigns local timing and sequence metadata for processing newly received input. A short-form packet supplies geometry and mode directly for bench testing of the same geometric solver; the JSON form carries the upstream status and activity fields.

5.3 Wristband firmware



(a) wristband assembly **(b)** IMU module **(c)** I2C multiplexer

Figure 9: Four-motor wristband assembly and representative sensing/control modules used for posture sensing and tactile rendering.

The wristband runs RT-Thread [17]. Its firmware is organized into five runtime responsibilities: motor-array initialization, vector-packet input, IMU pose estimation, cue solving, and haptic playback. The four vibration drivers are connected through an I2C multiplexer. During boot, the firmware scans each channel, initializes online motors in real-time playback mode, and keeps an online/offline record for later diagnosis.

Vector packets are received over BLE. The parser accepts JSON fields such as `mode`, `status`, `vector_active`, `vector_x_m`, `vector_y_m`, `vector_z_m`, and `distance_m`. If the status is not tracking, the vector is zeroed or treated as distant depending on the stop state. If any required vector component is missing while tracking is active, the packet is rejected rather than rendered.

The BMI270 provides a wrist-pose update every 20 ms. The haptic thread checks for new vectors every 10 ms while idle. Guidance vectors are sent once every 0.7 s (approximately 1.42 Hz). This interval was selected from the guidance experiments to give users time to distinguish successive tactile cues. It is the command update rate, not the motor vibration frequency or a general upper limit of skin sensitivity. A new vector arriving during a pulse pattern is handled after the current pattern finishes.

Source-record validation is assigned to the upstream cue core, while local sequence handling lets the haptic thread identify newly received input. The reported grasping trials used functioning posture compensation, which is required for grasping in the current configuration.

5.4 Runtime timing and cue patterns

The haptic solver initializes every cue as silent, then activates output only if the parsed vector is valid. In grasp mode, vectors shorter than 0.05 m are muted to represent arrival at the target. In avoid mode, a distance shorter than 0.05 m triggers a danger pattern with all four motors. Otherwise, the solver applies posture compensation, directional corner mapping, and distance gain.

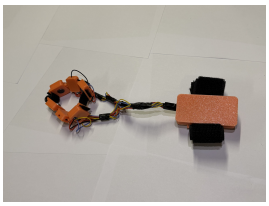
Directional haptic output uses one 140 ms pulse. Front, back, and danger conditions use separate pulse timing to encode depth and urgency. The design assigns semantic information to audio and directional information to haptics; the present evaluation does not separately quantify a reduction in repeated spoken instructions.



(a) stereo camera



(b) edge node



(c) wristband



(d) tabletop scene

Figure 10: Representative module-specific verification fixtures and physical devices used for the perception, edge-computing, haptic, and grasping tests.

6 Experimental Verification

6.1 Evaluation objectives

The evaluation verifies whether the prototype provides usable spatial evidence, whether the spatial-cue core prevents invalid evidence from reaching the actuators, and whether the integrated device supports repeated grasping and local obstacle avoidance. Device verification and trial archives are complemented by a separate participant study report comparing collision counts by visual status and device (Section 6.11).

6.2 Experimental platform and scenarios

The platform is the three-node prototype described in Section 3. Grasping trials use a tabletop scene with common objects; avoidance trials include stone bollards, bicycles, and pedestrians. The evidence includes device trials, offline perception evaluations, and sensor recordings with different logging coverage. These records are analyzed separately rather than treated as one fully synchronized experiment.

Table 5 summarizes the evaluation matrix. Each experiment below identifies its available records and measurement scope; condition ordering and participant allocation are not inferred where they are absent from the supplied records.

6.3 Runtime measurements

Table 4 separates latency, throughput, and configured scheduling values. The entries come from separate component runs and therefore do not form a synchronized end-to-end latency budget. The YOLO11s value is aggregate throughput over the 1,004-image detection evaluation; no YOLO11s single-frame latency is inferred from it. The LightStereo record is a different 12-frame run: 126.95 ms is LightStereo model inference only, whereas 228.39 ms is its complete export loop. The latter corresponds to 4.38 frames/s when executed sequentially.

The 0.7 s command interval (approximately 1.42 Hz) is a configured guidance-update schedule rather than a measured inference throughput, so it is reported with the firmware timing in Section 5. The camera entry is a capture-to-host frame-receipt interval and should not be interpreted as pure USB-link latency. Likewise, the Bluetooth values measure only the duration of the host-side write call; mechanical cue onset is evaluated independently from accelerometer records.

6.4 Module-level verification protocol

Table 5 defines the module-level experiments for the quantitative verification pass.

Table 4: Runtime and scheduling measurements from separate component runs. Each row identifies one metric; values are not summed as an end-to-end latency budget.

Component or stage	Metric	Statistic	Value	Measurement scope
Stereo-camera acquisition	Configured rate	Nominal	30 frames/s	Camera configuration
Camera capture to host frame receipt	Latency	Approximate	100 ms	Ends when the RK3588 host receives the frame; no internal transport/buffering breakdown
YOLO11s inference on a 2-TOPS device	Throughput	Mean	11.38 frames/s	1,004-image detection evaluation
LightStereo, 320×576	Inference latency	Mean	126.95 ms/frame	Model execution only, $N = 12$ frames
LightStereo, 320×576	Loop latency	Mean	228.39 ms/frame	Complete frame export loop, $N = 12$; equivalent to 4.38 frames/s
Cloud SAM3	Throughput	Observed	3.1 frames/s	Separate cloud component run
Cloud Depth Anything 3	Throughput	Observed	7.7 frames/s	Separate cloud component run
Direct decision	Latency	Median	5.3 μs	3,600 software decisions
Direct decision	Latency	P95	5.8 μs	Same run
Spatial-cue decision	Latency	Median	25.1 μs	3,600 software decisions
Spatial-cue decision	Latency	P95	36.5 μs	Same run
Bluetooth write call (BLE host path)	Call latency	Mean	0.316 ms	Host-side `write()` duration, 674 frames
Bluetooth write call (BLE host path)	Call latency	P95	0.377 ms	Not transmission or end-to-end cue latency
Qwen prefill	Token throughput	Observed range	25–35 tokens/s	Separate language-model run
Qwen decode	Token throughput	Observed range	12–15 tokens/s	Separate language-model run

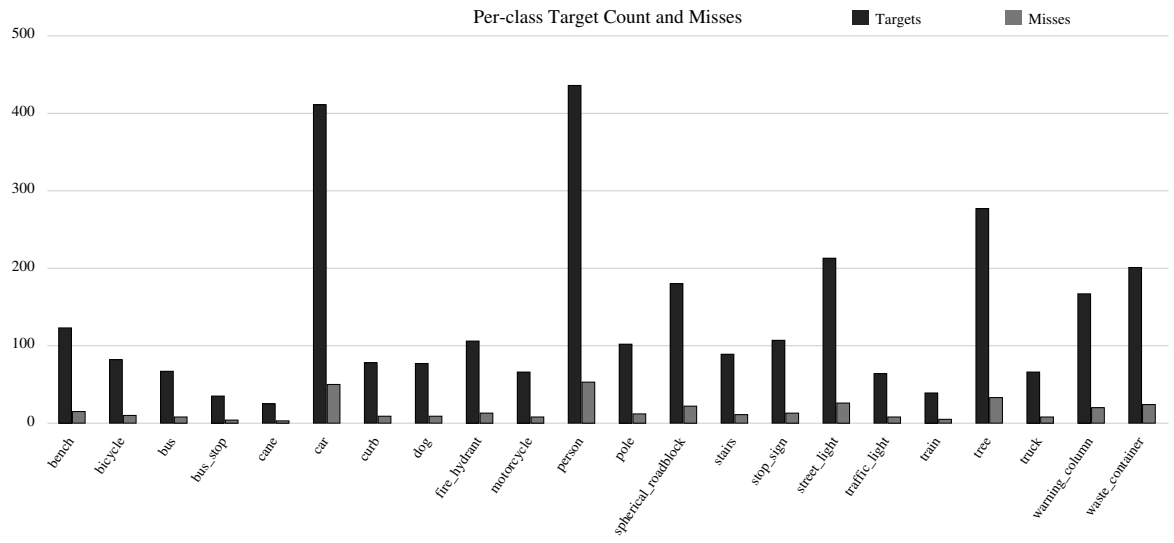
Detection is evaluated against labeled targets in the OOD test set. Depth estimates are compared with the measured reference distance. Absolute depth error is

$$e_z = |\hat{z} - z_{gt}|. \tag{17}$$

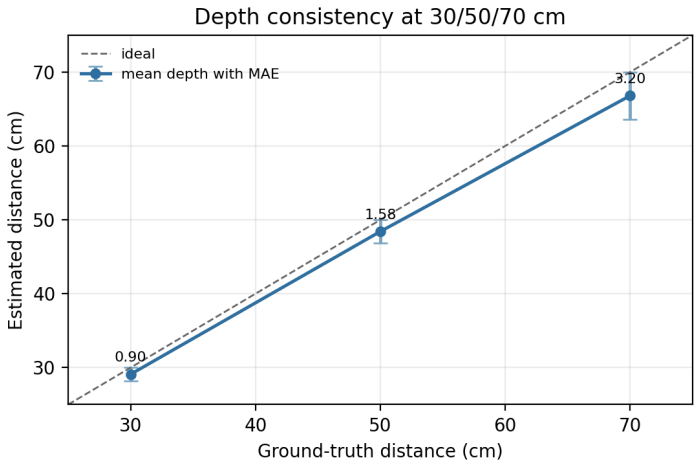
Haptic playback is checked with canonical vector packets covering directional, stop, and danger states. Posture compensation is required for successful grasping in the current prototype and was enabled in all grasping trials. No separate compensation-disabled trial series was conducted.

Table 5: Evaluation matrix for the wearable guidance prototype.

Test target	Main condition	Primary metrics
Object and hazard detection	Repeated indoor scenes with target variation, clutter, and partial occlusion	Precision, recall, miss rate, and inference delay
Metric depth and 3D vector formation	Known-distance planar and object targets with measured lateral and forward offsets	Depth MAE/RMSE, temporal variation, and vector completeness
Haptic rendering	Canonical direction packets and accelerometer onset records	Direction recognition, detected onset delay, and detection coverage
Cue-state safety	Logged replay with missing entity/depth, incomplete vectors, stale outputs, and mode changes	Valid-cue preservation, invalid-cue suppression, and false suppression
Integrated assistance	Repeated object grasping and local obstacle-avoidance runs	Completion rate, time, cue reversals, minimum clearance, and failure category



(a) targets and missed detections by class



(b) depth consistency at 30/50/70 cm

Figure 11: Perception test results. Panel (a) shows labeled targets and missed detections by class. Panel (b) compares depth estimates with the three reference distances.

Table 6: Module-level verification summary from the available records.

Module	Record	Result
Detection	1004 test images, 3011 targets	Precision 84.60%, recall 87.91%, F1 86.22%, miss rate 12.09%, throughput 11.38 FPS
Depth consistency	30/50/70 cm, 100 frames each	Mean depth 29.10/48.42/66.80 cm; MAE 0.90/1.58/3.20 cm; RMSE 0.90/1.58/3.23 cm
Cue-core replay	525 valid-cue records + 436 invalid active records	436/436 invalid active cues suppressed, 0 false suppression
Wrist direction pilot	two-participant pilot, 40 trials each	Strict exact recognition 63/80 (78.8%); relaxed adjacent-direction recognition 78/80 (97.5%)
BLE write path	COM11, 674 frames	100% write success; 0.316 ms mean write duration; 0.377 ms P95

Table 7: Placement-level descriptive errors relative to the exported ruler-based reference coordinates. Each of the 15 placement IDs at a nominal distance contributes one mean; the reported spread is the sample SD across placement means, not across repeated frames.

Nominal z (cm)	Placements n	Valid/all frames	Depth MAE (cm)	Vector MAE (cm)	Angle mean (deg)
30	15	733/750	0.56 ± 0.42	1.13 ± 0.38	4.81 ± 2.19
50	15	729/750	0.65 ± 0.28	1.66 ± 0.47	5.93 ± 2.93
70	15	730/750	0.83 ± 0.26	1.76 ± 0.30	6.48 ± 1.89

The detection test uses 1004 labeled images containing 3011 targets, with confidence threshold 0.25 and IoU threshold 0.5. Table 6 reports the updated module results. Figure 11(a) shows target counts and misses from the updated class-level export. The depth test uses 100 frames per distance and the median valid depth in a box-face ROI, with disparity scale 1.

6.5 Reference-based depth and relative-vector measurements

A separate measured placement dataset contains 2250 frames at nominal forward placements of 30, 50, and 70 cm, with 15 placement IDs and 750 frames per distance. Prediction and reference coordinates refer to corresponding physical hand and target points. We recompute the relative vectors as $\hat{\mathbf{v}} = \hat{\mathbf{p}}_t - \hat{\mathbf{p}}_h$ and $\mathbf{v}_{\text{ref}} = \mathbf{p}_{t,\text{ref}} - \mathbf{p}_{h,\text{ref}}$ using the exported reference coordinates. Vector error is $\|\hat{\mathbf{v}} - \mathbf{v}_{\text{ref}}\|_2$, and target-depth error is $|\hat{p}_{t,z} - p_{t,z,\text{ref}}|$. Vector MAE is the mean Euclidean error and vector RMSE is the square root of its mean square.

The reference coordinates were measured with a ruler relative to the midpoint of the line connecting the two cameras. Hand and target coordinates use matching axes, so their relative vector is obtained by subtraction.

Coordinates are finite in 2192/2250 frames; the other 58 frames are retained in the availability denominator and excluded from error averages. For angle error, we use the angle between the two relative vectors and require both norms to exceed 1 mm. This leaves 731, 729, and 730 angle samples at the three respective distances. To avoid treating repeated frames as independent replicates, each placement ID is first reduced to its mean error and the statistical unit is the placement ($n = 15$ per distance). Table 7 therefore reports placement-level mean±sample-SD values; the earlier frame-weighted means are not used in the abstract. These errors are relative to the supplied ruler-based reference coordinates and are separate from the scalar-depth consistency test. Ruler placement and reference-point localization introduce measurement uncertainty that is not separately quantified

Table 8: Cue-core replay event tally for the available active-window log.

Condition	Direct baseline	Proposed cue core
Valid cues emitted under no fault	525/525	525/525
Invalid active cues propagated	436/436	0/436
Invalid active cues suppressed	0/436	436/436
Observed false suppression	0	0

Table 9: Accelerometer-detected onset latency over 339 detected events out of 360 triggers. SD is the sample standard deviation; undetected events are excluded from latency statistics.

Interval	Mean (ms)	SD (ms)	Median (ms)	P95 (ms)
Receive to onset	58.68	36.77	51.00	92.75
Drive to onset	54.36	36.78	46.50	88.75

here.

6.6 Cue-core replay and validity testing

The spatial-cue core is evaluated with timestamped replay so that the direct-mapping baseline and the proposed contract receive identical perception records. Direct mapping renders a vector whenever a vector-like field is present. The proposed core additionally checks entity grounding, depth and vector completeness, mode consistency, activity, and freshness before rendering.

The available grasping trace contains 916 records, including 305 active records. A fixed 120-record active window was replayed with no fault and with injected missing-entity, missing-depth, partial-vector, frozen-output, and mode-switch conditions. Five predetermined seeds were used for paired replay, giving 60 comparison trials. The denominators in Table 8 are event counts from the exported replay tally: 525 valid-cue evaluation events and 436 invalid-active evaluation events. They are not obtained by multiplying the 916-record trace length, the 120-record window, or the number of seeds; those quantities describe the replay construction, whereas the tally counts the evaluated cue events after scenario filtering.

The replay confirms the deterministic validity contract on the available trace. It does not replace the module-level detection and depth experiments because replay isolates cue-state logic rather than perception accuracy.

6.7 Wristband pilot verification

Mechanical vibration onset was detected with an accelerometer. The export contains 360 triggers, of which 339 have a detected onset (94.2%); 13 are labeled no mechanical onset and eight sensor detection failure. Table 9 is recomputed from the recorded receive, drive, and detected-onset timestamps. Undetected events are counted separately and are not assigned zero latency. The results characterize accelerometer-detected onset, including the detection procedure's delay. They do not measure the user's perceptual response time. The export records a 2 kHz sampling rate and a 1 s detection window; the mounting, onset threshold, and synchronization implementation are not specified in the supplied acquisition description.

Two participants each completed 40 grasp-mode direction trials. Exact recognition was 63/80 (78.8%); allowing either adjacent 45-degree direction gave 78/80 (97.5%). The two scores describe

Table 10: Grasping outcomes under the confirmed 60 s timeout. Completion time is mean ± sample SD over successful trials only.

Measure	Result
Successful trials	37/40 (92.5%)
Completion time, successful trials	20.21 ± 7.99 s
Elapsed time, all trials	21.04 s
Trials with a logged wrong grasp	8/40

different levels of directional precision. This direction test is separate from the four-participant object-grasping experiment.

The host-side BLE write path remained stable in the updated test: 674 of 674 frames produced successful host write calls, the mean write-call duration was 0.316 ms, and the P95 duration was 0.377 ms. These values do not measure radio transmission or wristband reception latency. The distance-to-amplitude curve used by the firmware maps 0.05, 0.10, 0.20, 0.30, 0.50, 0.70, and 1.00 m to amplitudes 95, 91, 83, 76, 61, 46, and 23, respectively, with values clipped below the near threshold and above the far threshold.

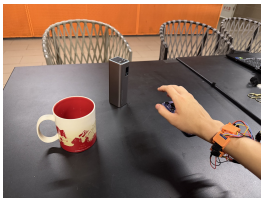
6.8 Integrated grasping and avoidance trials

Integrated trials evaluate whether the same cue interface can drive both reaching and local movement. In grasping, the operator starts with the hand at a fixed start area and receives non-visual guidance toward the requested object. A run is successful if the correct object is lifted within 60 s without visual inspection. The supplied trial export contains four participant IDs with ten trials each. It records 37/40 successful trials (92.5%); successful-trial completion time is 20.21 ± 7.99 s (mean ± sample SD), while mean elapsed time over all 40 trials, including failures, is 21.04 s.

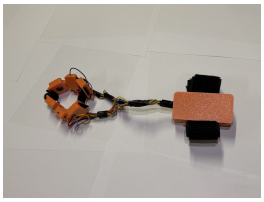
Failures occurred when the hand contacted another object too early or the target moved outside the camera view. The available notes do not assign a count to each cause. Eight trials included a wrong-grasp event, but some of these trials later succeeded. We therefore report intermediate wrong grasps separately from final task failures.



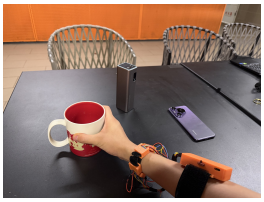
(a) tabletop target scene



(b) guided reaching



(c) haptic interface



(d) target grasped

Figure 12: Representative grasping-assistance process, including target scene, guided reaching, wristband feedback, and successful object retrieval.

Table 11: Raw-log coverage and motion statistics in two obstacle-avoidance archives, reported as mean ± sample SD across runs. Geographic samples are retained only as coverage indicators because their reported horizontal uncertainty is too large for metric clearance reconstruction. Acceleration uses each run's P95. Recording duration is distinct from task completion time.

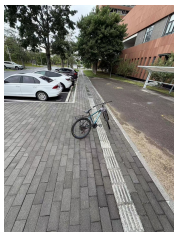
Scenario	Recordings	Log span (s)	Geo samples
Pedestrian obstacle	16	9.73 ± 2.33	5.31 ± 0.95
Stone bollard obstacle	15	11.96 ± 2.20	13.67 ± 2.38

Scenario	Roll SD (deg)	User accel. P95 (m/s^2)
Pedestrian obstacle	2.53 ± 1.17	1.68 ± 0.16
Stone bollard obstacle	3.62 ± 1.69	2.33 ± 0.23

In avoidance, the operator walks through a marked corridor while receiving hazard cues when applicable. A run is successful if the operator passes the obstacle without contact. For the complete-system and speech conditions, cue correctness (whether the cue indicated the obstacle side before closest approach) is treated as a separate diagnostic; it is not part of the shared success definition and is not applicable to the white-cane condition. Failure categories are collision/contact, late cue, wrong direction, and runtime stall when such labels are available.



(a) stone bollards



(b) bicycle



(c) pedestrian

Figure 13: Representative local obstacle-avoidance scenes used to visualize how the system guides local lateral movement around different hazards.

6.9 Trajectory context and obstacle-avoidance metrics

The sensor archives contain 16 pedestrian recordings and 15 stone-bollard recordings. These 31 recordings describe logging coverage, not the number of successes or attempts in the three-condition comparison. No trial-ID mapping between the two sources is assumed. For a measured path $\mathbf{q}_t = (x_t, z_t)$, obstacle center $\mathbf{o}$, and radius r_o, path length and clearance are defined as

$$L = \sum_{t=2}^{T} \|\mathbf{q}_t - \mathbf{q}_{t-1}\|_2 , \qquad C_{\min} = \min_t \left(\|\mathbf{q}_t - \mathbf{o}\|_2 - r_o\right). \tag{18}$$

The raw `geo_location.csv` files provide time stamps and coarse geographic samples, but the archived horizontal localization uncertainty (18–42 m across the two scenarios) is not sufficient for metric clearance calculation. The three-condition task, path, and clearance values below are reported results transcribed from the accompanying comparison report; they are not independently recomputed from the archived sensor logs or rasterized panels. The report is the provenance record for those aggregate values, while the present raw-log archive supports only coverage and motion

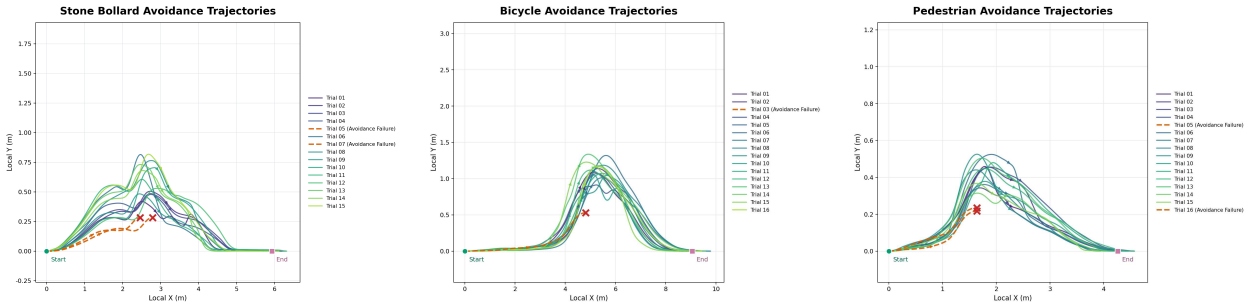
Table 12: Overall obstacle-avoidance results transcribed from the supplied comparison report. These are reported aggregate values; they are not recomputed from the archived sensor logs in Table 11.

Condition	Trials	Success (%)	Time (s)	Avg. clearance (m)	Min. clearance (m)
Complete system	47	89.4%	26.7	0.63	0.26
Speech only	47	80.9%	34.6	0.49	0.16
White cane	47	87.2%	35.4	0.55	0.18

Table 13: Scenario-level completion results transcribed from the supplied comparison report. These reported aggregates are separate from the archived sensor logs.

		Complete system		Speech only		White cane	
Scenario	Trials	Success (%)	Time (s)	Success (%)	Time (s)	Success (%)	Time (s)
Stone bollard	15	86.7	24.3	86.7	31.0	86.7	31.5
Pedestrian	16	87.5	28.7	75.0	37.1	87.5	38.0
Bicycle	16	93.8	27.0	81.3	35.5	87.5	36.5

summaries. The 31 sensor recordings are therefore not treated as the statistical unit for the 47-trial-per-condition comparison.



(a) stone bollards **(b)** bicycle **(c)** pedestrian

Figure 14: Qualitative overhead trajectory visualizations for stone-bollard, bicycle, and pedestrian obstacles. Each line represents one logged trial trajectory; the start, end, and contact markers identify recorded trial events. The panels provide scene and path context only and are not the source from which quantitative completion, path, or clearance values are recomputed.

6.10 Reported obstacle-avoidance comparison

The comparison report covers the complete glasses-plus-wristband system, speech-only guidance, and a white cane. Each condition comprises 47 reported trials across stone bollard (15), pedestrian (16), and bicycle (16) scenarios. Tables 12–14 transcribe the report's overall, task, and spatial-safety results. These aggregate values should not be interpreted as independently reproducible from the 31 archived sensor recordings in Table 11; those recordings are a separate source used only for coverage and motion statistics.

The complete system achieved 89.4% success with a mean time of 26.7 s, a reported path-averaged clearance of 0.63 m, and a minimum safe clearance of 0.26 m. Speech guidance achieved 80.9%, 34.6 s, 0.49 m, and 0.16 m, respectively; the white-cane condition achieved 87.2%, 35.4 s, 0.55 m, and 0.18 m. Thus, the reported mean times were 7.9 s and 8.7 s longer than for the

Table 14: Scenario-level spatial-safety results transcribed from the supplied comparison report. Clearances are distances to the obstacle boundary over successful trajectories in the valid interaction segment; they are not recomputed from the archived sensor logs.

Scenario	Condition	Avg. clearance (m)	Min. clearance (m)	Path length (m)	Max. offset (m)
Stone bollard	Complete system	0.52	0.30	6.30	0.81
	Speech only	0.43	0.22	6.42	0.64
	White cane	0.47	0.24	6.50	0.86
Pedestrian	Complete system	0.43	0.26	4.52	0.53
	Speech only	0.31	0.16	4.58	0.47
	White cane	0.36	0.18	4.65	0.58
Bicycle	Complete system	0.92	0.58	9.39	1.34
	Speech only	0.73	0.44	9.46	1.33
	White cane	0.82	0.47	9.58	1.35

complete system, while their minimum safe clearances were 0.10 m and 0.08 m smaller. Clearance statistics summarize successful trajectories in the valid obstacle-interaction segment; unsuccessful trials remain included in the success-rate denominator.

The 47-trial-per-condition report provides aggregate percentages and means rather than trial-level paired observations. Its time and clearance contrasts are therefore descriptive; participant-level variance cannot be recovered from these means. Visual-status comparisons are addressed using the separate collision-count report in Section 6.11.

Exploratory between-condition and within-scene analysis. The aggregate report nevertheless permits a transparent effect-size description. Relative to the white-cane condition, the complete system increased the reported overall success rate by 2.2 percentage points, reduced mean completion time by 8.7 s (24.6%), increased path-averaged clearance by 0.08 m, and increased minimum safe clearance by 0.08 m. Relative to speech-only guidance, the corresponding differences were +8.5 percentage points, −7.9 s (22.8%), +0.14 m, and +0.10 m. These contrasts are practical differences in the reported aggregates, not inferential tests. The scenario-level table shows the same pattern in two scenes: complete-system and white-cane success rates are identical for the stone-bollard scene (86.7% vs. 86.7%) and the pedestrian scene (87.5% vs. 87.5%), whereas the complete system is higher in the bicycle scene (93.8% vs. 87.5%). Thus, the available scene summaries are consistent with comparable completion performance between the proposed system and the cane, together with faster completion and larger reported clearances for the proposed system.

6.11 Visual-status groups and paired device comparisons

A supplementary participant study report describes 100 volunteers recruited at Nanfang Hospital, Southern Medical University: 50 blind participants (no light perception or light perception only) and 50 participants with low vision (best-corrected visual acuity below 0.3, with residual vision beyond light perception). This study is analyzed separately from the 47-trial-per-condition comparison above; its participant counts and collision outcomes are not pooled with those trial aggregates. Mean age was 65.2 ± 6.4 years in the blind group and 61.4 ± 6.8 years in the low-vision group. The report gives an age difference of $t(98) = 2.87$, $p = 0.005$, and no significant difference in sex distribution ($p > 0.05$). Age was included as a covariate in the overall model. Numerical test results in this subsection follow the supplied participant report.

Design and statistical criteria. The report describes a 2×2 mixed design with visual status as a between-participant factor and device (smart glasses versus white cane) as a within-participant factor. Within each visual-status group, half of the participants used the cane first and half used the glasses first, with an interval of at least 30 min. The primary outcome was the number of torso or head contacts with obstacles on a standardized indoor route, recorded by trained independent assessors blinded to the study hypotheses. Contacts made by the cane itself are not this outcome.

The participant is the statistical unit. The analysis strategy first examines the group-by-device interaction in a mixed design, with age adjustment through a linear mixed-effects model, followed by within-group device contrasts. Let $D_i = C_{i,\text{cane}} - C_{i,\text{glasses}}$ be a participant's paired collision-count difference. Positive values favor the glasses. Paired superiority is assessed with $t = \bar{D}/(s_D/\sqrt{n})$, $n-1$ degrees of freedom, and a two-sided significance level of 0.05. The paired standardized effect is $d_z = \bar{D}/s_D$. For equivalence in the low-vision group, the report specifies a margin of $\Delta = 1.5$ collisions, attributed to expert consensus and pilot data, and two one-sided tests (TOST), each at $\alpha = 0.05$: equivalence requires the 90% confidence interval for $\bar{D}$ to lie entirely within $[-1.5, 1.5]$. A nonsignificant superiority test alone does not establish equivalence.

Table 15: Collision counts in the supplementary participant report. Values are mean ± SD; difference SD is the SD of within-participant cane-minus-glasses differences.

Visual status	n	White cane	Smart glasses	Paired difference
Blind	50	12.54 ± 2.21	4.46 ± 1.35	8.08 ± 1.85
Low vision	50	5.12 ± 1.14	4.98 ± 1.08	0.14 ± 0.65

Within-group device contrasts. In the blind group, the glasses were associated with 8.08 fewer collisions than the cane (means 4.46 versus 12.54). The participant report gives $t(49) = 30.92$, $p < 0.001$, and Cohen's $d = 4.37$ for the paired comparison. These reported results support a device advantage for this group on the indoor collision outcome.

In the low-vision group, the paired difference was 0.14 collisions (reported 90% CI $[-0.01, 0.29]$). The report gives lower- and upper-bound TOST statistics of $t(49) = 17.85$ and $t(49) = -16.22$, respectively, with both one-sided $p < 0.001$. The reported interval lies entirely within the ± 1.5-collision margin, supporting equivalence of the two devices for this outcome under that margin. This conclusion concerns devices within the low-vision group; it does not establish equivalence between blind and low-vision participants using the glasses.

Between-group interpretation. The reported mean device benefit was 8.08 collisions in the blind group and 0.14 in the low-vision group, a descriptive difference in benefit of 7.94 collisions. The report gives an age-adjusted group-by-device interaction of $F(1, 97) = 412.5$, $p < 0.001$, with partial $\eta^2 = 0.81$. It also reports nonsignificant device-order and order-by-device effects (both $p > 0.10$). The interaction indicates that the reported device benefit differs by visual status. These are source-reported model results; their verification requirements are discussed under Limitations. The study addresses group differences and paired device effects, not a continuous linear correlation between visual acuity and benefit.

7 Discussion

The central design choice is to make the perception-to-feedback translation explicit. A detector can fail, a depth map can be missing, and a mode switch can leave an old vector in memory. The

upstream spatial-cue core checks task, entity, mode, metric relation, and freshness; the wristband converts the resulting vector into posture-compensated motor commands. Replay evaluates the upstream checks on shared perception records, and the wristband measurements and task trials characterize feedback execution and integrated operation.

The visual-to-vector formulation also clarifies what the system does and does not solve. In grasp mode, the cue is a local hand-to-object displacement, not an autonomous manipulation policy. In avoid mode, the cue is a short-range hazard warning, not a global navigation planner. This distinction is important for a wearable BLV assistant because local cues can support immediate reaching or sidestepping while leaving route choice and mobility strategy under user control.

Wrist rotation changes the relationship between the four fixed motors and the intended locally referenced direction. The BMI270-based roll estimate and eight-way quantized avoid mapping provide an embedded mechanism for compensating this rotation. In the current prototype, disabling compensation prevents successful grasping, so compensation was treated as an operational prerequisite and retained in every grasping trial. A separate compensation-disabled trial series was not performed; no numerical ablation effect is claimed. The implementation remains inspectable through its vector parsing and motor-command calculations.

The supplementary participant report suggests that device benefit depends on visual status: collision reduction favors the glasses in the blind group, whereas the device contrast meets the specified equivalence criterion in the low-vision group. This supports evaluating assistance against each user's functional needs. It does not establish a visual-acuity threshold, a causal mechanism involving residual vision, or a general recommendation to replace cane training.

7.1 Limitations

The participant comparison used a standardized indoor obstacle route and assessed short-term performance. Further evaluation should examine outdoor environments and adaptation during extended use. Collision counts were the primary participant outcome; future studies can add walking speed, gait stability, cognitive workload, and perceived safety to characterize the user experience more fully. Equivalence in the low-vision group applies to the specified collision outcome and the ± 1.5-collision margin.

The separate placement dataset supports descriptive vector errors relative to ruler-based coordinates, with unquantified reference-placement uncertainty. Accelerometer-detected onset includes detection delay. The 47-trial-per-condition obstacle report supplies aggregate success, time, path, and clearance values that cannot be independently recomputed from the archived sensor logs; participant allocation and trial-level dispersion are still required for inference on those outcomes. The system provides local obstacle warning rather than route planning and is not a replacement for established mobility skills.

Detection and depth tests measure perception accuracy. Cue-core replay checks how missing or inconsistent input is handled. Checking that a vector is complete does not establish that its object label or position is correct.

Future work should extend participant evaluation to longitudinal and outdoor settings, make participant-level analyses reproducible, improve stale-cue invalidation latency, add explicit confidence calibration for depth and detection, and integrate richer trajectory logging. The spatial-cue contract can also be extended to stair/curb warning or shelf search.

8 Conclusion

This paper presents a wearable non-visual guidance system for object grasping and local obstacle avoidance. The system integrates smart glasses, edge perception, and a four-motor wristband, but its main contribution is the spatial-cue core between perception and feedback. By representing each cue as task intent, grounded entity, mode, metric relation, and validity state, the system can explain why a cue is active, suppressed, or complete before the downstream audio and haptic renderers act.

The formulation connects depth-supported visual detections to relative vectors and posture-compensated haptic patterns. Reference-based measurements, cue-core replay, accelerometer onset records, and task trials characterize the implemented pipeline. The supplementary 100-participant report adds a mixed-design comparison by visual status: glasses reduce reported collision counts relative to the white cane in the blind group, whereas the device contrast in the low-vision group meets the specified equivalence criterion. The reported age-adjusted interaction supports a benefit that depends on visual status. These findings motivate assistance tailored to functional needs; they do not establish equivalence between the two visual-status groups or justify universal replacement of the white cane. The architecture separates upstream cue validation from wristband feedback execution through a compact vector interface.

Ethics approval and consent to participate

All participants provided informed consent before taking part in the reported engineering experiments. The work involved prototype evaluation rather than clinical intervention, and no participant-identifying information is included in this manuscript.

Consent for publication

All authors consent to submission and publication of this manuscript and its accompanying figures. The authors request publication through the journal's standard non-open-access/subscription route, subject to the journal's publication policy.

Data and materials availability

The code, analysis scripts, selected data exports, and supporting materials will be made available through the authors' GitHub repository. The repository link will be provided and maintained with the final journal version so that the materials can be accessed and updated through that repository.

Funding

This work was funded by the three corresponding authors: Wenxuan Ma, Fan Wang, and Yang Gao.

Competing interests

The authors declare that there is no conflict of interest among them in this paper.

Authors' contributions

Chenghang Li designed the glasses-side electronics, designed and implemented the device appearance, wrote the main-controller and glasses-side software, and performed testing, tuning, and model deployment. Yirui Yuan designed and implemented the wristband, wrote the relevant main-controller software, and performed testing, tuning, and model deployment. Shurui Zhang wrote the main-controller and vision software and performed model deployment, testing, and tuning. The corresponding authors Wenxuan Ma, Fan Wang, and Yang Gao provided funding, supervised the work, guided the literature review, and reviewed and verified the manuscript and its analyses. All authors read and approved the final manuscript.

References

[1] Michael Ahn et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 556–568, Auckland, New Zealand, 2022. PMLR.

[2] Jinqiang Bai, Zhaoxiang Liu, Yimin Lin, Ye Li, Shiguo Lian, and Dijun Liu. Wearable travel aid for environment perception and navigation of visually impaired people, 2019. `arXiv:1904.13037`.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report, 2025. `arXiv:2511.21631`.

[4] Anthony Brohan et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183, Atlanta, GA, USA, 2023. PMLR.

[5] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, et al. SAM 3: Segment anything with concepts, 2025. `arXiv:2511.16719`.

[6] Dima Damen et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision*, pages 720–736, Munich, Germany, 2018. Springer.

[7] Dima Damen et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022.

[8] Danny Driess et al. Palm-e: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 8469–8488, Honolulu, HI, USA, 2023. PMLR.

[9] Espressif Systems. ESP-IDF programming guide for ESP32-C3, 2026. Accessed 2026-07-15. URL: `https://docs.espressif.com/projects/esp-idf/en/latest/esp32c3/`.

[10] Marcin Furtak, Florian Pätzold, Tim Kietzmann, Silke M. Kärcher, and Peter König. Helping blind people grasp: Enhancing a tactile bracelet with an automated hand navigation system, 2025. `arXiv:2504.16502`.

[11] Kristen Grauman et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1895–1914, New Orleans, LA, USA, 2022. IEEE/CVF.

[12] Xianda Guo, Chenming Zhang, Youmin Zhang, Wenzhao Zheng, Dujun Nie, Matteo Poggi, and Long Chen. LightStereo: Channel boost is all you need for efficient 2d cost aggregation, 2024. arXiv:2406.19833.

[13] Chutian Jiang, Emily Kuang, and Mingming Fan. How can haptic feedback assist people with blind and low vision (BLV): A systematic literature review, 2024. arXiv:2412.19105.

[14] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, 2023. URL: https://github.com/ultralytics/ultralytics.

[15] Haotong Lin et al. Depth anything 3: Recovering the visual space from any views, 2025. arXiv:2511.10647.

[16] Xiaoan Liu, DaeHo Lee, Eric J. Gonzalez, Mar Gonzalez-Franco, and Ryo Suzuki. Visionclaw: Always-on ai agents through smart glasses, 2026. arXiv:2604.03486.

[17] RT-Thread Community. RT-Thread documentation, 2026. URL: https://www.rt-thread.io/document/site/.

[18] Ligao Ruan et al. A multimodal assistive system for product localization and retrieval for people who are blind or have low vision, 2026. arXiv:2601.12486.

[19] Yuanming Shi, Kai Yang, Tao Jiang, Jun Zhang, and Khaled B. Letaief. Communication-efficient edge ai: Algorithms and systems. *IEEE Communications Surveys & Tutorials*, 22(4):2167–2191, 2020. doi:10.1109/COMST.2020.3007787.

[20] Pengzhan Sun et al. Visual intention grounding for egocentric assistants, 2025. arXiv:2504.13621.

[21] Issatay Tokmurziyev et al. Llm-glasses: Genai-driven glasses with haptic feedback for navigation of visually impaired people, 2025. arXiv:2503.16475.

[22] Sicheng Yang et al. Egocentric co-pilot: Web-native smart-glasses agents for assistive egocentric ai, 2026. arXiv:2603.01104.