

Multi-Modal Range Fusion for Agile Quadruped Navigation in Dense Vegetation

Farah Idris • Hao Chen • Kai Wen Lim

Robotics and Control Laboratory, Nanyang Technological University, Singapore

PUBLISHED

July 24, 2026

REGISTERED

July 25, 2026

RECORD TYPE

Preprint

ABSTRACT

Quadruped robots are expected to move quickly through forests, parks, and overgrown terrain, yet LiDAR range sensing degrades precisely there: foliage produces premature returns, thin stems and branches fall between beams, and occlusion hides the rigid structure that constrains motion. This paper presents a range-fusion method that combines LiDAR and stereo-vision range estimates with weights conditioned on where the robot is predicted to move. A configuration-driven local planner, in the sense of AgniNav, supplies the robot's collision envelope and candidate trajectories; a first-order model of velocity tracking converts them into a motion-predicted occupancy field over the sensing grid. Per-cell fusion weights combine this field with modality-specific noise, motion-induced latency error, and a vegetation degradation model whose density parameter is estimated online from LiDAR transmittance. The method is evaluated on a Unitree Go2 carrying a stereo rig and a 32-beam LiDAR, in an Isaac Lab synthetic forest suite with four vegetation density classes and on a 600 m park course through shrub and sapling cover. Compared with LiDAR-only sensing, it reduces range estimate error from 0.99 m to 0.94 m, raises mean speed from 2.07 m/s to 2.31 m/s, and raises obstacle-free success from 84.6 to 90.7 percent, while mean obstacle clearance decreases from 18.1 cm to 16.8 cm. Uniform-weight fusion reaches only 84.9 percent success, and removing any single component lowers success. A failure taxonomy covers dense sapling cover and head-height branches.

INDEX TERMS quadruped navigation, LiDAR-stereo fusion, dense vegetation, motion-conditioned perception, configuration-driven local planning, field robotics

KAGG NOTE Range estimates matter most where the robot is about to go, and the planner that decides where the robot goes is the right source of that information.

01 Introduction

Quadruped robots are now fast and robust enough to be deployed in forests, parks, orchards, and overgrown field sites, where learned locomotion controllers traverse roots, mud, and uneven ground that defeat wheeled platforms [1, 2, 3]. Navigation in such places, however, remains limited less by locomotion than by range perception. A spinning LiDAR, the default exteroceptive sensor of field-deployed legged systems [4, 5], behaves poorly in dense vegetation: leaves and fine twigs produce premature returns that make free space look occupied, thin stems and branches fall between the vertical beams of a sparse scanner, and the near-field foliage occludes the rigid structure that actually constrains the motion of the body. These failure modes are most severe exactly where agile navigation matters most, namely when the robot must move quickly through narrow gaps between saplings while maintaining a safe margin to rigid obstacles.

Stereo vision is complementary in this setting. A stereo rig samples the scene densely in angle and therefore resolves thin vertical and overhanging structure that a 32-beam scanner misses, and it updates faster than a typical spinning LiDAR. Its depth error, however, grows quadratically with range, and repetitive, low-contrast, or self-similar foliage texture degrades disparity matching [6, 7, 8]. Neither modality is uniformly bet-

ter, and the relative quality of the two changes with range, viewing direction, and local vegetation density.

The standard response is to fuse the two range sources with weights derived from their noise models, uniformly over the sensed volume. This paper argues that uniform weighting optimizes the wrong objective for agile navigation. A local planner does not consume the whole point cloud equally; the decisions that determine collision and speed depend on range estimates along the corridor the robot will actually sweep during the next one to two seconds. Errors in directions the robot will never approach are largely irrelevant, while errors, latency, and over-smoothing inside the predicted corridor translate directly into late braking or unnecessary slowdowns. The planner itself already knows where the robot is going: a configuration-driven local planner in the sense of AgniNav [9] describes the robot through an explicit body envelope and motion limits and selects a commanded trajectory every cycle. The gap this paper addresses is that this motion information is not used upstream, in the range estimator.

We therefore propose *control-aligned range fusion*. The planner's configuration and its most recent commanded trajectories are propagated through a model of how the quadruped actually tracks velocity commands, yielding a motion-predicted occupancy field that assigns each sensing direction a relevance equal to the probability, discounted by time, that the robot's body will pass near it. This field, together with an online

estimate of vegetation density and a per-modality degradation model, sets the fusion weights, so that estimation accuracy and responsiveness are concentrated where the robot will go.

The contributions of this paper are as follows.

1. A formulation of range fusion for navigation as minimization of a *task-weighted* range error, in which the weight of each sensing direction is the predicted relevance of that direction to the robot's realized motion rather than a uniform prior.
2. A motion-predicted occupancy field computed from a configuration-driven planner's body envelope, its candidate trajectories, and a first-order model of quadruped velocity tracking, and a fusion rule whose weights depend on this field through both a spatial relevance factor and a motion-induced latency penalty.
3. A vegetation degradation model, estimated online from LiDAR transmittance statistics under a Beer-Lambert attenuation assumption, that predicts foliage pre-emption of LiDAR returns and the degradation of stereo matching.
4. An evaluation on a Unitree Go2 with a stereo rig and a 32-beam LiDAR, in an Isaac Lab synthetic forest suite with four vegetation density classes and on a 600 m park course through shrub and sapling cover, reporting range error, obstacle clearance, speed maintenance, success, ablations, and a failure taxonomy.

02 Related Work

2.1 Learned and perceptive legged locomotion

Reinforcement learning in simulation has become the dominant route to robust quadruped locomotion. Hwangbo et al. trained policies with learned actuator models and transferred agile skills to ANYmal [1], and Lee et al. showed that a proprioceptive teacher-student policy can traverse challenging natural terrain [2]. Miki et al. extended this to perceptive locomotion by fusing proprioception with a noisy height map through a learned belief encoder, explicitly reasoning about when exteroception is unreliable, including in vegetation where the height map reports foliage as solid [3]. Rudin et al. made such training fast through massively parallel GPU simulation [10], and Kumar et al. introduced rapid motor adaptation, in which an adaptation module infers environment extrinsics online [11]. Egocentric-vision policies have since enabled stair and gap traversal from a single depth camera [12], cross-modal transformers combine proprioception with depth for obstacle-rich terrain [13], and multi-modal delay randomization addresses the latency between vision and proprioception in the wild [14]. Parkour systems push agility further [15, 16, 17]. These works treat the locomotion controller as the consumer of perception; our method instead takes a capable locomotion policy as given and asks how upstream range estimation should be shaped by the motion it will produce. The idea of reasoning about exteroceptive reliability in [3] is closest in spirit to our degradation model, but it acts at the level of footholds, whereas we model degradation of range along sensing rays for local planning.

2.2 Local navigation for legged robots in the field

Several systems couple perception and navigation for legged robots. Rudin et al. learned locomotion and local navigation end-to-end [18], Hoeller et al. learned a compact state representation for navigation in cluttered and dynamic scenes

[19], and Fu et al. coupled vision with proprioception so that a legged robot can detect collisions and terrain changes that vision alone misses [20]. GPU elevation mapping provides fast 2.5D terrain representations for locomotion and navigation [21], and large field deployments such as CERBERUS and NeBula's Autonomous Spot demonstrate long-range legged autonomy with LiDAR-centric mapping [4, 5]. In vegetated terrain specifically, traversability learning has shown that geometry alone misclassifies compliant vegetation: BADGR learns event probabilities from self-supervision [22], WayFAST predicts traversability from wheel-terrain interaction [23], and Wild Visual Navigation learns traversability online from pre-trained visual features [24, 25]. High-speed flight through forests has also been learned from depth input with simulated privileged experts [26]. These methods largely concern *what* is traversable. Our work is complementary: we target the accuracy of range to rigid structure and allocate that accuracy according to predicted motion, and a learned traversability layer could be placed on top of the fused estimate.

2.3 Configuration-driven and cross-embodiment local planning

AgniNav [9] is the planning framework on which our method builds. It formulates local planning around an abstract robot configuration: a shared four-parameter collision envelope (collision-relevant height, front length, rear length, and half width) together with sensing and velocity limits. The height parameter conditions the perception front end, and the planar parameters configure collision checking, so that a single planning stack can run unchanged across embodiments once their configuration is declared. Two properties of AgniNav are central here. First, the explicit envelope makes the robot's swept volume computable for any candidate trajectory, which is exactly what is needed to predict which sensing directions the body will approach. Second, the planner's interface is geometric rather than learned end-to-end, so an upstream estimator can be modified without retraining the planner. AgniNav itself treats the range input as given; our contribution is to feed its configuration and trajectory selection back into range estimation. We therefore use AgniNav "in the sense of" its configuration-driven structure rather than claiming improvements to its planning algorithm. CeRLP [27] pursues a related cross-embodiment goal by converting heterogeneous visual observations into a unified geometric scan representation with scale-corrected depth and height-adaptive abstraction; like AgniNav it standardizes the interface between perception and planning, and our fusion can be read as a way to populate such an interface from two modalities under degradation. Data-driven generalist policies such as GNM [28] share knowledge across robots through standardized actions instead of explicit envelopes and are therefore less directly suited to the swept-volume reasoning we require.

Classical local planners provide the trajectory-sampling and safety structure we adopt. The dynamic window approach evaluates admissible velocity pairs under acceleration and braking constraints [29], and fast aerial planners such as Fast-Planner and EGO-Planner optimize trajectories against distance fields at high speed [30, 31]. Xiao et al. survey how learning has been combined with such classical planners [32]. None of these planners modulates the fidelity of its own perception input; they assume a map of uniform quality.

2.4 Range sensing, mapping, and fusion under degradation

Occupancy grids [33] and signed distance fields [34] remain the standard representations for local planning, and LiDAR-inertial odometry such as LIO-SAM and FAST-LIO2 provides the motion compensation and ego-motion estimates on which they rely [35, 36]. These pipelines treat each LiDAR return as a sample of a surface, an assumption violated by foliage. Stereo matching, from semi-global matching [6] to learned cost volumes [7] and recurrent refinement [8], yields dense depth whose accuracy depends on texture and range. Depth completion methods fuse sparse LiDAR with images to recover dense depth [37, 38], and adverse-weather fusion has shown that learned, entropy-steered fusion can survive degradation of individual sensors [39]. Uncertainty-aware deep perception distinguishes aleatoric from epistemic uncertainty [40], a distinction we mirror by separating per-modality noise from vegetation-induced outlier processes. All of these methods weight modalities by sensor quality alone. To our knowledge, they do not condition fusion on the downstream controller’s predicted motion, which is the central difference of the present work.

2.5 Simulation for legged navigation

GPU-parallel simulators [41] and the Orbit framework, which has since evolved into Isaac Lab [42], make it practical to evaluate perception and navigation across many procedurally generated scenes. Domain randomization [43] motivates our use of multiple seeds and photorealistic foliage materials, although, as discussed in Section 6, simulated foliage remains an imperfect proxy for real vegetation.

03 Problem Formulation

3.1 Robot, configuration, and motion

The robot state at time t is the planar pose $\xi_t = (\mathbf{p}_t, \theta_t) \in SE(2)$ together with the body-frame velocity $\mathbf{u}_t = (v_t, \omega_t)$, where v_t is forward speed and ω_t is yaw rate. Following Agni-Nav [9], the robot is described by a configuration

$$\mathbf{c} = (h, l_f, l_r, w, v_{\max}, \omega_{\max}, a_{\max}), \quad (1)$$

where h is the collision-relevant height, l_f and l_r are the front and rear lengths of the body envelope measured from the body origin, w is its half width, $v_{\max}$ and $\omega_{\max}$ are the speed and yaw-rate limits, and $a_{\max}$ is the admissible longitudinal deceleration. The body envelope at pose ξ is the box $\mathcal{B}(\xi; \mathbf{c}) \subset \mathbb{R}^3$ of length $l_f + l_r$, width $2w$, and height h . The planner issues a velocity command $\mathbf{u}_t^{\text{cmd}}$; the realized velocity $\mathbf{u}_t$ differs from it because of the locomotion controller’s tracking dynamics, slip, and disturbances.

3.2 Sensing grid and measurements

Range is represented on a robot-centric spherical grid with N_θ azimuth bins and N_ϕ elevation bands. Each cell $i \in \mathcal{I}$ has unit direction $\mathbf{d}_i$ and an unknown true range $r_i^* \in (0, r_{\max}]$ to the first *rigid* (collision-relevant) surface along $\mathbf{d}_i$; leaves and compliant foliage are not rigid surfaces in this definition. At each cycle the LiDAR provides a motion-compensated measurement z_i^L , taken as the nearest return in cell i , and the stereo pipeline provides z_i^S , the median of valid depths whose back-projected rays fall in cell i . Either measurement may be missing. Azimuth bins are grouped into N_k sectors $k = 1, \dots, N_k$, and $k(i)$ denotes the sector of cell i . Each sector carries an unknown vegetation density $\rho_k \geq 0$, interpreted as a foliage area density with units of m^{-1} .

3.3 Task-weighted estimation objective

Let $\hat{r}_i$ denote the fused range estimate. A uniform estimator minimizes $\mathbb{E} [\sum_i (\hat{r}_i - r_i^*)^2]$. For navigation, the cost of an error in cell i depends on whether the robot will come close to the structure along $\mathbf{d}_i$. We define the relevance $M_i \in [0, 1]$ of cell i as the time-discounted probability that the body envelope, under the robot’s *realized* motion over a horizon T_h , passes within a kernel distance of the ray segment of cell i (Section 4.4). The estimation problem is

$$\min_{\hat{\mathbf{r}}} \mathbb{E} \left[\sum_{i \in \mathcal{I}} (\epsilon + M_i) (\hat{r}_i - r_i^*)^2 \right], \quad (2)$$

where $\epsilon > 0$ is a small floor that keeps every cell weakly observed so that the planner can react if its plan changes. The planner then selects a trajectory from a candidate set using the fused estimate and its uncertainty. Because M_i depends on the trajectory the planner selects and the planner depends on $\hat{\mathbf{r}}$, the two are coupled; Section 4 breaks the loop by computing M_i from the previous planning cycle.

04 Method

4.1 System overview and timing

The stack runs four loops. The LiDAR produces scans at 10 Hz and the stereo pipeline produces disparity at 15 Hz. Fusion and planning run at 10 Hz, triggered by each LiDAR scan and using the most recent stereo frame after motion compensation with LiDAR-inertial odometry [36]. The planner outputs a velocity command that is interpolated and passed to the Go2 locomotion policy, which runs with a 20 ms control period (50 Hz). Each fusion cycle performs: (i) propagation of the previous fused estimate to the current pose; (ii) per-sector vegetation density update; (iii) computation of the motion-predicted occupancy field from the previous cycle’s planner output; (iv) per-cell weighted fusion with a disagreement test; and (v) planning on the fused estimate.

4.2 Sensor and degradation models

LiDAR. In the absence of vegetation, a LiDAR return in cell i is modeled as $z_i^L = r_i^* + \eta^L$ with $\eta^L \sim \mathcal{N}(0, \sigma_{L,0}^2)$. In vegetation, a beam traveling toward a rigid surface at range r may be pre-empted by foliage. Under a Beer–Lambert attenuation model, the probability that the beam passes a path of length r through foliage of density ρ without interception is $\exp(-\beta \rho r)$, where β is a dimensionless projection factor that accounts for leaf orientation and beam footprint. The LiDAR likelihood is therefore the mixture

$$p(z_i^L | r_i^*) = (1 - \pi_i) \mathcal{N}(z_i^L; r_i^*, \sigma_{L,i}^2) + \pi_i \frac{\mathbb{I}[0 < z_i^L < r_i^*]}{r_i^*}, \quad (3)$$

with pre-emption probability $\pi_i = 1 - \exp(-\beta \rho_{k(i)} r_i^*)$ and in-mixture variance $\sigma_{L,i}^2 = \sigma_{L,0}^2 + c_L \rho_{k(i)}$, where c_L captures the additional range spread caused by partial occlusion of the beam footprint. The uniform term models a premature return at an arbitrary depth inside the foliage layer.

Stereo. For focal length f (pixels), baseline B , and disparity noise σ_d (pixels), first-order error propagation of $z = fB/d$ gives the standard depth standard deviation

$$\sigma_{S,i} = \frac{(r_i^*)^2}{fB} \sigma_d(\rho_{k(i)}), \quad \sigma_d(\rho) = \sigma_{d,0} (1 + c_S \rho), \quad (4)$$

where c_S models the increase of matching ambiguity in self-similar foliage. Stereo does not suffer pre-emption in the same way, because a disparity is assigned to the surface that dominates the matching window, but it has an outlier probability π_i^S that we set from the fraction of pixels in cell i failing the left-right consistency check.

4.3 Online vegetation density estimation

The density ρ_k is estimated per sector from LiDAR transmittance. Let $\bar{r}_i$ be the prior range of cell i , obtained by transporting the previous fused estimate to the current pose with odometry. A LiDAR return in cell i is labeled *short* if $z_i^L < \bar{r}_i - \delta_i$, with $\delta_i = 3\sqrt{\sigma_{L,i}^2 + \bar{\sigma}_i^2}$ and $\bar{\sigma}_i$ the prior standard deviation, and *passing* if it is consistent with $\bar{r}_i$. Let n_k^s and n_k^p be the numbers of short and passing returns in sector k , and $\bar{\ell}_k$ the mean prior range of those cells, which serves as the mean path length through the foliage layer. The empirical transmittance $\hat{T}_k = n_k^p / (n_k^p + n_k^s)$ and the Beer-Lambert relation give the instantaneous estimate

$$\tilde{\rho}_k = -\frac{\ln \max(\hat{T}_k, T_{\min})}{\beta \bar{\ell}_k}, \quad (5)$$

where $T_{\min}$ bounds the estimate when nearly all returns are short. The estimate is smoothed with an exponential filter,

$$\rho_k \leftarrow (1 - \alpha_k) \rho_k + \alpha_k \tilde{\rho}_k, \quad \alpha_k = \alpha_0 \frac{n_k^p + n_k^s}{n_k^p + n_k^s + n_0}, \quad (6)$$

so that sectors with few returns adapt slowly; $\alpha_0 \in (0, 1]$ is the maximum gain and n_0 a pseudo-count. Equation (5) is the maximum-likelihood estimate of ρ for a binomial count of passing returns with success probability $\exp(-\beta \rho \bar{\ell}_k)$. Because the labels depend on the prior, density is estimated against the fused map rather than against LiDAR alone, which prevents a dense foliage layer from being absorbed into the prior and thereby declared transparent.

4.4 Motion-predicted occupancy field

Realized-motion model. The locomotion policy tracks velocity commands with lag and noise. We model the realized velocity with a first-order tracking model,

$$\dot{v} = \frac{v^{\text{cmd}} - v}{\tau_v} + w_v, \quad \dot{\omega} = \frac{\omega^{\text{cmd}} - \omega}{\tau_\omega} + w_\omega, \quad (7)$$

where τ_v and τ_ω are tracking time constants and w_v, w_ω are zero-mean white noise with intensities q_v, q_ω that account for slip and gait-induced oscillation. The pose evolves by unicycle kinematics $\dot{\mathbf{p}} = v [\cos \theta, \sin \theta]^\top$, $\dot{\theta} = \omega$. Equation (7) is discretized with step $\Delta t = 20$ ms, matching the locomotion control period, and the realized speed is clipped to $[0, v_{\max}]$.

Candidate mixture. The planner of the previous cycle evaluated a set of candidate commands $\{\mathbf{u}^{(j)}\}_{j=1}^J$ with costs C_j (Section 4.6). Because the next selection may differ from the

last one, the predicted motion is a mixture over the N_c lowest-cost candidates with Boltzmann weights

$$\lambda_j = \frac{\exp(-C_j/T_c)}{\sum_{j'=1}^{N_c} \exp(-C_{j'}/T_c)}, \quad (8)$$

where T_c is a temperature that sets how strongly the field concentrates on the selected trajectory.

Field. For each candidate j we draw K rollouts $\{\xi_t^{(j,m)}\}$ of Eq. (7) from the current state estimate. Let $\mathcal{S}_i = \{s \mathbf{d}_i : 0 \leq s \leq \bar{r}_i + \Delta_r\}$ be the ray segment of cell i extended by a margin Δ_r , and let $\text{dist}(\mathcal{B}, \mathcal{S})$ denote the Euclidean distance between the body envelope and the segment. The motion-predicted occupancy field is

$$M_i = \sum_{j=1}^{N_c} \lambda_j \frac{1}{K} \sum_{m=1}^K \max_{0 \leq t \leq T_h} e^{-t/\tau_h} \cdot \kappa(\text{dist}(\mathcal{B}(\xi_t^{(j,m)}); \mathbf{c}), \mathcal{S}_i), \quad (9)$$

with the Gaussian kernel $\kappa(d) = \exp(-d^2/2\ell_\kappa^2)$, discount time constant τ_h , and kernel width ℓ_κ . The maximum over time records whether the body ever approaches the ray; the discount makes near-term interactions dominate. Because $\mathcal{B}$ is three-dimensional with height h , elevation bands above the envelope receive low relevance unless the body approaches them closely, while a branch at the height of the body or sensor mast falls inside the envelope and receives high relevance. M_i is the quantity in Eq. (2). Alongside M_i we record the expected closing speed toward each cell,

$$\bar{v}_i = \sum_j \lambda_j \frac{1}{K} \sum_m \max_{0 \leq t \leq T_h} [\mathbf{v}_t^{(j,m)} \cdot \mathbf{d}_i]_+, \quad (10)$$

where $\mathbf{v}_t^{(j,m)}$ is the planar velocity of rollout (j, m) expressed in the current body frame and $[\cdot]_+ = \max(\cdot, 0)$.

4.5 Control-aligned fusion

Inlier responsibilities. For each modality $m \in \{L, S\}$, the posterior probability that z_i^m is an inlier, evaluated at the prior $\bar{r}_i$, is

$$\gamma_i^L = \frac{(1 - \pi_i) \mathcal{N}(z_i^L; \bar{r}_i, \sigma_{L,i}^2 + \bar{\sigma}_i^2)}{(1 - \pi_i) \mathcal{N}(z_i^L; \bar{r}_i, \sigma_{L,i}^2 + \bar{\sigma}_i^2) + \pi_i \mathbb{I}[z_i^L < \bar{r}_i]/\bar{r}_i}, \quad (11)$$

with π_i evaluated at $\bar{r}_i$, and analogously γ_i^S with the stereo outlier probability π_i^S and a uniform outlier density on $(0, r_{\max}]$.

Motion-conditioned weights. Each modality has a measurement latency δ_m , the time between the physical sampling of the scene and the availability of the measurement to the fusion step. During this interval the robot closes on cell i at a speed that the field predicts, so stale measurements carry an additional error of order $\delta_m \bar{v}_i$ even after nominal motion compensation, because compensation uses the estimated rather than the realized motion. The per-cell, per-modality weight is

$$w_i^m = \frac{\gamma_i^m}{\sigma_{m,i}^2 + (\zeta \delta_m \bar{v}_i)^2}, \quad (12)$$

where $\zeta \in [0, 1]$ is the fraction of latency-induced displacement that remains after compensation. The weights therefore

depend on the predicted realized motion through $\bar{v}_i$: when the robot is about to move quickly toward a cell, the lower-latency modality is favored for that cell; when it is not, both modalities are weighted by their static noise.

Field-level estimate. The fused range field minimizes the quadratic surrogate of Eq. (2),

$$J(\mathbf{r}) = \sum_i (\epsilon + M_i) \sum_{m \in \{L, S\}} w_i^m (r_i - z_i^m)^2 + \lambda_p \sum_i \frac{(r_i - \bar{r}_i)^2}{\bar{\sigma}_i^2 + q_r \Delta T} + \lambda_s \sum_{(i, i') \in \mathcal{E}} (r_i - r_{i'})^2, \quad (13)$$

where missing measurements contribute no data term, $q_r \Delta T$ inflates the prior over the cycle period $\Delta T = 0.1$ s, $\mathcal{E}$ is the set of 4-neighbor pairs on the grid, and λ_p, λ_s weight the temporal prior and the spatial smoothness prior. The relevance factor $\epsilon + M_i$ multiplies only the data term. In cells the robot is about to approach, the data dominate the temporal prior and the smoothness term, so the estimate responds within one cycle to a newly observed obstacle and is not blurred across a gap; in irrelevant cells the estimate leans on the prior and on its neighbors, which suppresses noise-driven replanning. Equation (13) is a sparse, symmetric positive-definite quadratic. It is minimized with conjugate gradients warm-started at $\bar{\mathbf{r}}$, with a fixed iteration budget. The per-cell posterior variance is approximated by the diagonal of the inverse Hessian, using

$$\hat{\sigma}_i^2 \approx \left[(\epsilon + M_i) \sum_m w_i^m + \lambda_p (\bar{\sigma}_i^2 + q_r \Delta T)^{-1} + 2\lambda_s |\mathcal{N}_i| \right]^{-1}, \quad (14)$$

where $|\mathcal{N}_i|$ is the number of neighbors of cell i . This is a lower bound on the marginal variance and is used only for the planner's confidence margin.

Disagreement test. Quadratic fusion averages two inconsistent hypotheses, which is unsafe when one sensor sees a thin obstacle that the other misses, for example a branch between two LiDAR beams. When both measurements are present and $|z_i^L - z_i^S| > \chi \sqrt{\sigma_{L,i}^2 + \sigma_{S,i}^2}$, the cell is treated as bimodal. Let z_i^{near} be the nearer measurement and P_i^{near} its normalized weight $w_i^{\text{near}} / (w_i^L + w_i^S)$. If $M_i \geq M_{\text{thr}}$ and $P_i^{\text{near}} \geq p_{\text{min}}$, the data term of the farther measurement is removed for this cycle, so that the nearer hypothesis is retained; otherwise the cell is fused normally. The test is risk-sensitive only where the robot is about to go, which avoids the general conservatism of always taking the minimum range.

4.6 Planning on the fused estimate

The planner follows the configuration-driven structure of Agni-Nav [9]. Candidate commands are constant-curvature arcs $\mathbf{u}^{(j)} = (v^{(j)}, \omega^{(j)})$ on a grid over $[0, v_{\text{max}}] \times [-\omega_{\text{max}}, \omega_{\text{max}}]$, rolled out over the horizon T_h . For collision checking, each cell is converted into a lower-confidence obstacle point at range $\hat{r}_i - k_\sigma \hat{\sigma}_i$, and a candidate is admissible only if the swept envelope $\mathcal{B}(\xi_t; \mathbf{c})$ keeps a margin d_0 to all such points. Each admissible candidate receives the cost

$$C_j = w_g \Delta g_j + w_c \sum_t \phi(d_{j,t}^{\text{clr}}) + w_v (v_{\text{max}} - v^{(j)})^2 + w_s \|\mathbf{u}^{(j)} - \mathbf{u}_{t-1}^{\text{cmd}}\|^2, \quad (15)$$

where Δg_j is the remaining distance to the local goal at the end of the rollout, $d_{j,t}^{\text{clr}}$ is the envelope clearance at time t , $\phi(d) = \max(0, d_c - d)^2$ penalizes clearance below a comfort distance d_c , and the last term penalizes command changes. The selected forward speed is finally limited by the stopping-distance constraint

$$v^{\text{cmd}} \leq \sqrt{2 a_{\text{max}} \max(0, d^{\text{free}} - d_0)}, \quad (16)$$

where d^{free} is the free distance along the selected arc. Because $\hat{\sigma}_i$ shrinks where M_i is large, the planner obtains tighter confidence bounds exactly along its own corridor, which permits higher speed without reducing the margin d_0 . The full candidate set and costs are passed to the next cycle to form Eq. (8).

4.7 Implementation parameters

The Go2 envelope including the sensor payload uses $l_f = 0.40$ m, $l_r = 0.30$ m, $w = 0.16$ m, and $h = 0.50$ m. Motion limits are $v_{\text{max}} = 3.0$ m/s, $\omega_{\text{max}} = 2.0$ rad/s, and $a_{\text{max}} = 2.5$ m/s²; v_{max} is set below the platform's unloaded top speed to account for the sensor payload. The planner horizon is $T_h = 2.0$ s with a 9×21 grid of candidate commands, $N_c = 5$, and $K = 16$ rollouts per candidate. The sensing grid has $N_\theta = 360$ azimuth bins and $N_\phi = 16$ elevation bands, with $r_{\text{max}} = 30$ m for planning; LiDAR returns beyond this range are used only for odometry. The remaining parameters ($\beta, c_L, c_S, \tau_v, \tau_\omega, \tau_h, \ell_K, \zeta, \lambda_p, \lambda_s, \chi, M_{\text{thr}}, p_{\text{min}}, k_\sigma$) were set once on held-out simulated scenes and kept fixed for all experiments.

05 Experiments

5.1 Platform and setup

All experiments use a Unitree Go2 quadruped carrying a stereo rig and a 32-beam LiDAR. The simulated evaluation uses an Isaac Lab [42] synthetic forest suite with four vegetation density classes and physically based rendering (PBR) materials for foliage, comprising 20 scenes evaluated with 5 seeds each and 30 trials per scene-seed configuration. The real-world evaluation uses a 600 m park course through shrub and sapling cover, with 8 runs per density class. Density class 4 corresponds to dense sapling cover.

5.2 Baselines and ablations

All methods use the same planner (Section 4.6), locomotion policy, and parameters; only the range estimate supplied to the planner differs.

- **LiDAR only:** Eq. (13) with the stereo data term removed and nominal LiDAR noise.
- **Stereo only:** Eq. (13) with the LiDAR data term removed and nominal stereo noise.
- **Uniform-weight fusion (no control alignment):** inverse-variance fusion of both modalities with fixed nominal noise, $M_i \equiv 1$, no latency term, and no vegetation degradation model; this is the conventional sensor-quality-only fusion.
- **Ours:** the full method of Section 4.

The ablations remove one component at a time from the full method: the vegetation degradation model ($\rho_k \equiv 0$), the motion-conditioned weights ($M_i \equiv 1$ and $\bar{v}_i \equiv 0$), or the stereo branch.

5.3 Metrics

Range estimate error is the mean absolute difference, in meters, between the fused range and the range to the nearest rigid surface along the evaluated cells. *Obstacle clearance* is the mean, over trials, of the minimum distance between the body envelope and rigid obstacles, in centimeters. *Mean speed* is the average forward speed over a trial in m/s and measures speed maintenance. *Obstacle-free success* is the percentage of trials that reach the goal without contact with a rigid obstacle.

5.4 Main comparison

Table 1 reports the main comparison. The full method obtains the lowest range estimate error, 0.94 m, compared with 0.99 m for LiDAR only, 1.08 m for stereo only, and 1.03 m for uniform-weight fusion, which are relative reductions of about 5%, 13%, and 9%. It also attains the highest mean speed, 2.31 m/s, which is 0.24 m/s (about 12%) above LiDAR only and uniform-weight fusion (both 2.07 m/s) and 0.13 m/s above stereo only (2.18 m/s). Obstacle-free success rises to 90.7%, from 84.6% for LiDAR only and 84.9% for uniform-weight fusion, gains of 6.1 and 5.8 percentage points; stereo only reaches 76.5%.

Obstacle clearance requires a more careful reading. The full method passes rigid obstacles with a mean minimum clearance of 16.8 cm, lower than LiDAR only (18.1 cm), uniform-weight fusion (18.6 cm), and stereo only (18.9 cm). The method does not reduce the planner margin d_0 ; rather, its tighter confidence bounds in the predicted corridor allow the planner to accept narrower gaps and to keep speed through them, and the baselines, whose range fields are inflated by premature foliage returns or by stereo noise, keep larger distances by detouring or slowing. Lower clearance with higher success is consistent with this interpretation, but it also means that the method operates closer to rigid structure, so any residual range bias in the corridor has less room to be absorbed. We regard the 1.3 to 2.1 cm reduction in mean clearance as a real cost that a deployment should weigh against the gains in success and speed.

The comparison between LiDAR only and uniform-weight fusion is also informative. Adding stereo with sensor-quality-only weights increases range error from 0.99 m to 1.03 m, leaves mean speed unchanged at 2.07 m/s, and changes success only from 84.6% to 84.9%. Fusion as such therefore does not account for the gains; the improvement appears only when fusion is aligned with predicted motion and with the vegetation model.

5.5 Ablation

Table 2 removes one component at a time. Removing the stereo branch has the largest effect, raising error to 1.18 m and lowering success to 68.4%, a drop of 22.3 percentage points. Removing the motion-conditioned weights raises error to 1.06 m and lowers success to 80.1% (10.6 points below the full method). Removing the vegetation degradation model raises error to 1.08 m and lowers success to 82.6% (8.1 points below).

Two observations deserve caution. First, each ablated variant has a lower success rate than the uniform-weight fusion baseline (84.9%), and the variant without the stereo branch is also worse than LiDAR only in both error (1.18 m versus 0.99 m) and success (68.4% versus 84.6%). The components are therefore not additive: a partial system can be worse than no alignment at all. A plausible explanation, which the present data do

not isolate, is that the components were tuned jointly. Without stereo, the degradation model discounts short LiDAR returns in dense sectors with no second modality to corroborate them, so some true stems are discounted as foliage. Without motion conditioning, the degradation model downweights LiDAR uniformly across dense sectors, including the near-field corridor where LiDAR returns from stems are accurate. Second, the ablation reports range error and success only; the effect of each component on clearance and speed is not part of the present record.

5.6 Failure taxonomy

Table 3 summarizes the recorded failure cases, which we group into three categories.

F1: Occlusion-induced track loss. In dense sapling cover (class 4), the LiDAR-only baseline lost track in 27% of trials. A likely mechanism, consistent with the degradation model of Section 4.2, is that premature foliage returns surround the robot and repeatedly overwrite the transported prior, so that the free corridor disappears from the range field. The proposed fusion held track in this class with 1.3 m error. Read as a range error, this figure is well above the pooled 0.94 m, which indicates that class 4 remains the hardest regime and that holding track does not imply accurate ranging there.

F2: Thin obstacles between LiDAR beams. On one park run, the robot approached a fallen branch at head height. The 32-beam LiDAR's vertical spacing left the branch unsampled until the robot was close, whereas the stereo rig flagged it 0.5 s before the first LiDAR return. This is the situation the disagreement test is designed for: a branch inside the envelope height h lies in cells of high relevance M_i , where the nearer stereo hypothesis is retained instead of being averaged with the farther LiDAR return. The robot stopped with a braking margin of 21 cm. For scale, at the mean speed of 2.31 m/s the 0.5 s lead corresponds to roughly 1.2 m of travel. We interpret the case as a near miss: a LiDAR-only system would have received its first return 0.5 s later, and the 21 cm margin suggests that little of this lead time was spare.

F3: Misallocated fusion accuracy. In class 4, uniform-weight fusion shows 34% higher range error than the proposed fusion. Uniform weighting spends its fidelity evenly over the sensed volume, including dense foliage the robot will never enter, while the relevant corridor receives no special treatment. This is the clearest evidence in the record that the benefit of control alignment is concentrated where the robot actually goes, and that it is pronounced in the densest cover.

REPRODUCIBILITY NOTE

All methods share the planner, locomotion policy, sensor configuration, and parameters; only the range field supplied to the planner differs. The tables report aggregate figures over the evaluation protocol; a separate breakdown by simulation and park course, and confidence intervals, are not part of the present record.

Table 1: Main comparison on the Unitree Go2 (synthetic forest suite and park course). Lower is better for range estimate error; higher is better for mean speed and obstacle-free success. Obstacle clearance is reported without a preferred direction; see text.

Method	Range estimate error (m)	Obstacle clearance (cm)	Mean speed (m/s)	Obstacle-free success (%)
LiDAR only	0.99	18.1	2.07	84.6
Stereo only	1.08	18.9	2.18	76.5
Uniform-weight fusion (no control alignment)	1.03	18.6	2.07	84.9
Ours	0.94	16.8	2.31	90.7

Table 2: Ablation of the full method. Each row removes one component.

Variant	Range estimate error (m)	Obstacle-free success (%)
Ours (control-aligned fusion)	0.94	90.7
w/o vegetation degradation model	1.08	82.6
w/o motion-conditioned weights	1.06	80.1
w/o stereo branch	1.18	68.4

06 Limitations

Several limitations qualify the conclusions. First, the reported tables aggregate the synthetic forest suite and the park course, and the record does not include a separate breakdown by domain or by density class beyond the class-4 failure cases, nor confidence intervals; differences of a few hundredths of a meter in range error, such as 0.94 m versus 0.99 m, should be interpreted with this in mind. Second, the method operates with lower mean obstacle clearance than every baseline, and the safety of this trade-off rests on the calibration of the confidence bound $\hat{r}_i - k_\sigma \hat{\sigma}_i$, whose variance in Eq. (14) is a lower bound. Third, the motion-predicted field relies on a first-order tracking model, Eq. (7), that ignores gait-phase effects, foot slip on wet litter, and the pitching of the body, which shifts the sensors; large deviations between predicted and realized motion would place relevance on the wrong cells. Fourth, the Beer-Lambert density estimate assumes homogeneous foliage within a sector and uses the fused prior to label short returns, so a persistent bias in the prior can bias ρ_k . Fifth, the definition of rigid surfaces in simulation is a modeling choice; real vegetation includes stems that are partially compliant, and the method does not reason about traversability of compliant obstacles, which learned traversability estimators address [24, 25]. Sixth, the ablation shows that partial versions of the method can underperform the simpler uniform baseline, which makes the system sensitive to component failure, for example a degraded stereo camera in rain. Finally, all experiments use one platform, one sensor configuration, and one planner; the configuration-driven formulation suggests that the method should transfer across embodiments, but this has not been tested.

07 Conclusion

This paper presented control-aligned range fusion for agile quadruped navigation in dense vegetation. The method combines LiDAR and stereo range with weights that depend on a motion-predicted occupancy field, derived from a configuration-driven planner’s body envelope and candidate trajectories, and on an online vegetation degradation model. On a Unitree Go2 in a synthetic forest suite and a park course, it reduced range estimate error to 0.94 m, raised mean speed to 2.31 m/s, and raised obstacle-free success to 90.7%, compared with 0.99 m, 2.07 m/s, and 84.6% for LiDAR only, at the cost of a lower mean obstacle clearance of 16.8 cm. Uniform-

weight fusion did not reproduce these gains, and the recorded failure cases indicate that the benefit is concentrated in dense cover and in thin obstacles that a sparse LiDAR misses. Future work will report per-domain and per-class breakdowns with confidence intervals, replace the first-order tracking model with a learned model of realized motion, couple the fusion with learned traversability of compliant vegetation, and evaluate transfer across embodiments by varying the planner configuration.

References

- [1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning Agile and Dynamic Motor Skills for Legged Robots. *Science Robotics*, 2019. arXiv:1901.08652.
- [2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. *Science Robotics*, 2020. arXiv:2010.11251.
- [3] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Robust Perceptive Locomotion for Quadrupedal Robots in the Wild. *Science Robotics*, 2022. arXiv:2201.08117.
- [4] M. Tranzatto et al.. CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge. *Field Robotics*, 2022. arXiv:2201.07067.
- [5] A. Bouman et al.. Autonomous Spot: Long-Range Autonomous Exploration of Extreme Environments with Legged Locomotion. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. arXiv:2010.09259.
- [6] H. Hirschmüller. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328-341, 2008. doi:10.1109/TPAMI.2007.1166.
- [7] J.-R. Chang and Y.-S. Chen. Pyramid Stereo Matching Network. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. arXiv:1803.08669.
- [8] L. Lipson, Z. Teed, and J. Deng. RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. *International Conference on 3D Vision (3DV)*, 2021. arXiv:2109.07547.
- [9] T. Zang, S. Cheng, H. Huang, S. Wang, and W. Zhang. Agni-Nav: Configuration-Driven Cross-Embodiment Local Planning for Robot Navigation. arXiv:2606.10903, 2026.
- [10] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. *Conference on Robot Learning (CoRL)*, 2021. arXiv:2109.11978.
- [11] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. *Robotics: Science and Systems (RSS)*, 2021. arXiv:2107.04034.
- [12] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. *Conference on Robot Learning (CoRL)*, 2022. arXiv:2211.07638.

Table 3: Failure taxonomy derived from the recorded failure cases.

Category	Recorded case	Observed behavior
F1: Occlusion-induced track loss	Dense sapling cover (class 4)	LiDAR only lost track in 27% of trials; the proposed fusion held track with 1.3 m error.
F2: Sparse vertical sampling of thin overhanging obstacles	Fallen branch at head height on one park run	Stereo flagged the branch 0.5 s before the LiDAR return; braking margin of 21 cm.
F3: Misallocated fusion accuracy	Uniform-weight fusion in class 4	34% higher error than the proposed fusion in class 4.

- [13] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning Vision-Guided Quadrupedal Locomotion End-to-End with Cross-Modal Transformers. International Conference on Learning Representations (ICLR), 2022. arXiv:2107.03996.
- [14] C. S. Imai et al.. Vision-Guided Quadrupedal Locomotion in the Wild with Multi-Modal Delay Randomization. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022. arXiv:2109.14549.
- [15] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot Parkour Learning. Conference on Robot Learning (CoRL), 2023. arXiv:2309.05665.
- [16] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. arXiv:2309.14341, 2023.
- [17] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. Science Robotics, 2024. arXiv:2306.14874.
- [18] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022. arXiv:2209.12827.
- [19] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter. Learning a State Representation and Navigation in Cluttered and Dynamic Environments. IEEE Robotics and Automation Letters, 2021. arXiv:2103.04351.
- [20] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak. Coupling Vision and Proprioception for Navigation of Legged Robots. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. arXiv:2112.02094.
- [21] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter. Elevation Mapping for Locomotion and Navigation using GPU. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022. arXiv:2204.12876.
- [22] G. Kahn, P. Abbeel, and S. Levine. BADGR: An Autonomous Self-Supervised Learning-Based Navigation System. IEEE Robotics and Automation Letters, 2021. arXiv:2002.05700.
- [23] M. V. Gasparino et al.. WayFAST: Navigation with Predictive Traversability in the Field. IEEE Robotics and Automation Letters, 2022. arXiv:2203.12071.
- [24] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter. Fast Traversability Estimation for Wild Visual Navigation. Robotics: Science and Systems (RSS), 2023. arXiv:2305.08510.
- [25] M. Mattamala et al.. Wild Visual Navigation: Fast Traversability Learning via Pre-Trained Models and Online Self-Supervision. arXiv:2404.07110, 2024.
- [26] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. Learning High-Speed Flight in the Wild. Science Robotics, 2021. arXiv:2110.05113.
- [27] H. Xi, M. Tan, X. Zhang, S. Cheng, S. Wang, Y. Gu, X. Shen, and W. Zhang. CeRLP: A Cross-embodiment Robot Local Planning Framework for Visual Navigation. arXiv:2603.19602, 2026.
- [28] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot. IEEE International Conference on Robotics and Automation (ICRA), 2023. arXiv:2210.03370.
- [29] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. IEEE Robotics and Automation Magazine, 4(1):23-33, 1997. doi:10.1109/100.580977.
- [30] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen. Robust and Efficient Quadrotor Trajectory Generation for Fast Autonomous Flight. IEEE Robotics and Automation Letters, 2019. arXiv:1907.01531.
- [31] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao. EGO-Planner: An ESDF-free Gradient-based Local Planner for Quadrotors. IEEE Robotics and Automation Letters, 2021. arXiv:2008.08835.
- [32] X. Xiao, B. Liu, G. Warnell, and P. Stone. Motion Planning and Control for Mobile Robot Navigation Using Machine Learning: a Survey. Autonomous Robots, 2022. arXiv:2011.13112.
- [33] A. Elfes. Using Occupancy Grids for Mobile Robot Perception and Navigation. Computer, 22(6):46-57, 1989. doi:10.1109/2.30720.
- [34] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017. arXiv:1611.03631.
- [35] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020. arXiv:2007.00258.
- [36] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. IEEE Transactions on Robotics, 2022. arXiv:2107.06829.
- [37] F. Ma and S. Karaman. Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. IEEE International Conference on Robotics and Automation (ICRA), 2018. arXiv:1709.07492.
- [38] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity Invariant CNNs. International Conference on 3D Vision (3DV), 2017. arXiv:1708.06500.
- [39] M. Bijelic et al.. Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. arXiv:1902.08913.
- [40] A. Kendall and Y. Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? Advances in Neural Information Processing Systems (NeurIPS), 2017. arXiv:1703.04977.
- [41] V. Makoviychuk et al.. Isaac Gym: High Performance GPU-Based Physics Simulation for Robot Learning. NeurIPS Datasets and Benchmarks Track, 2021. arXiv:2108.10470.
- [42] M. Mittal et al.. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. IEEE Robotics and Automation Letters, 2023. arXiv:2301.04195.

- [43] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017. arXiv:1703.06907.