Localization and Mapping with 4D mmWave Radar

Ray Huang
CMU RI, PhD student





Motivation

SLAM in adverse environmental conditions



Thermal Camera: [Zhao et al. TP-TIO]

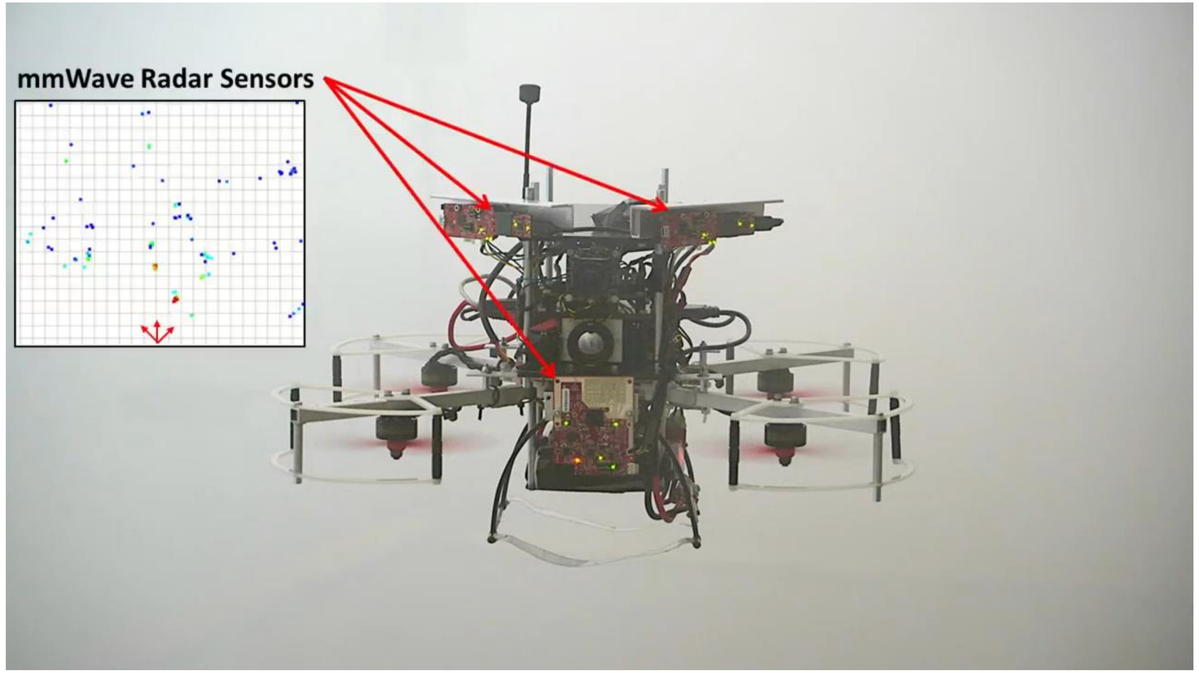


Rotating Radar : [Hong et al. RadarSLAM]

Motivation

SLAM in adverse environmental conditions

- Do not rely on external lighting
- Sensing in smoke, fog, rain, snow

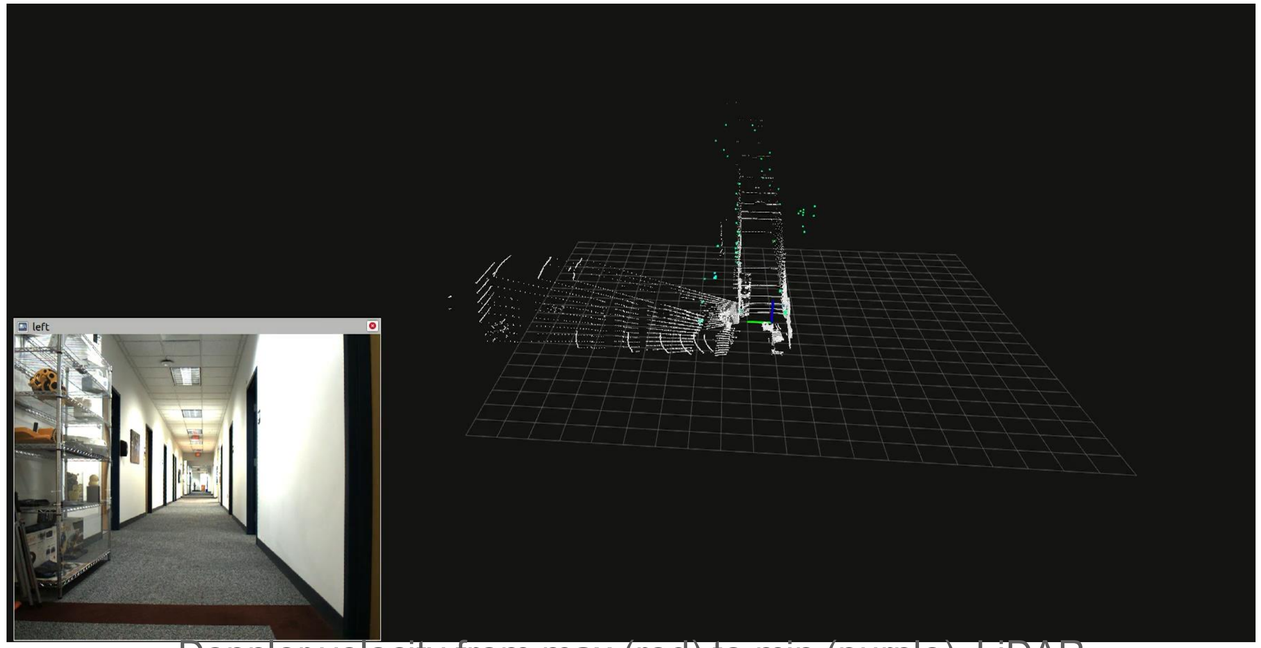


Imaging Radar [Doer et al. X-RIO]

Motivation

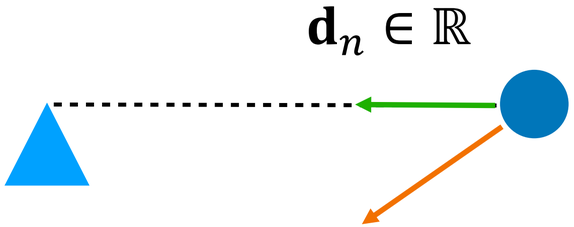
mmWave Imaging Radar

- **Properties:**
  - Sensing in adverse environment condition
  - Doppler velocity $\mathbf{d}_n \in \mathbb{R}$
  - Light weight
  - Low cost, easy to manufacture



Doppler velocity from max (red) to min (purple), LiDAR (white)

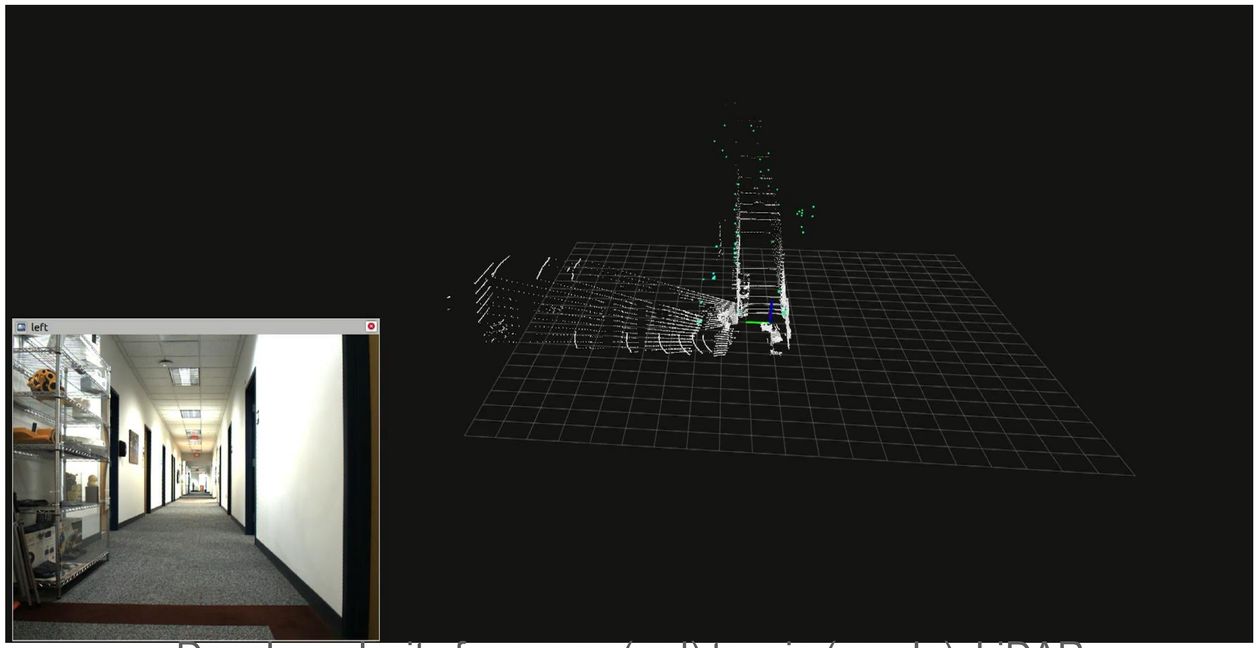




Motivation

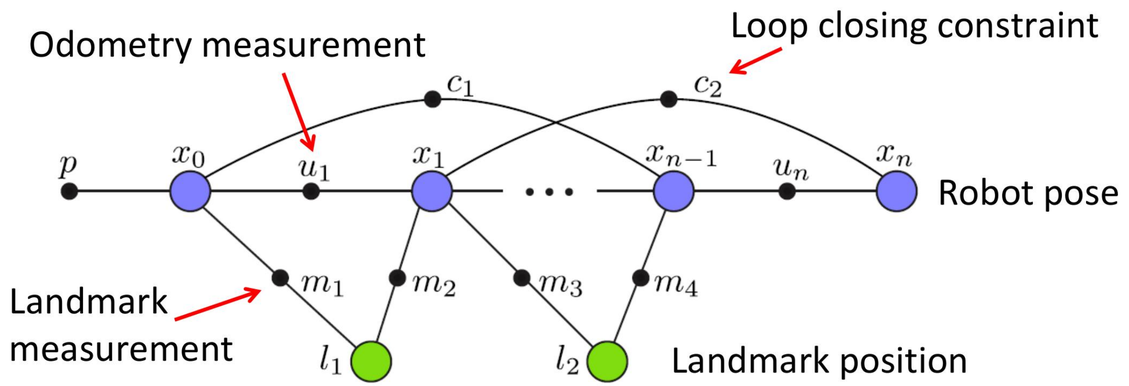
mmWave Imaging Radar

- **Challenges:**
  - Noisy signal response
  - Sparse measurement after signal processing
  - Discretized measurements
  - Discrepancy of resolution and FOV in elevation and azimuth dimension



Doppler velocity from max (red) to min (purple), LiDAR (white)

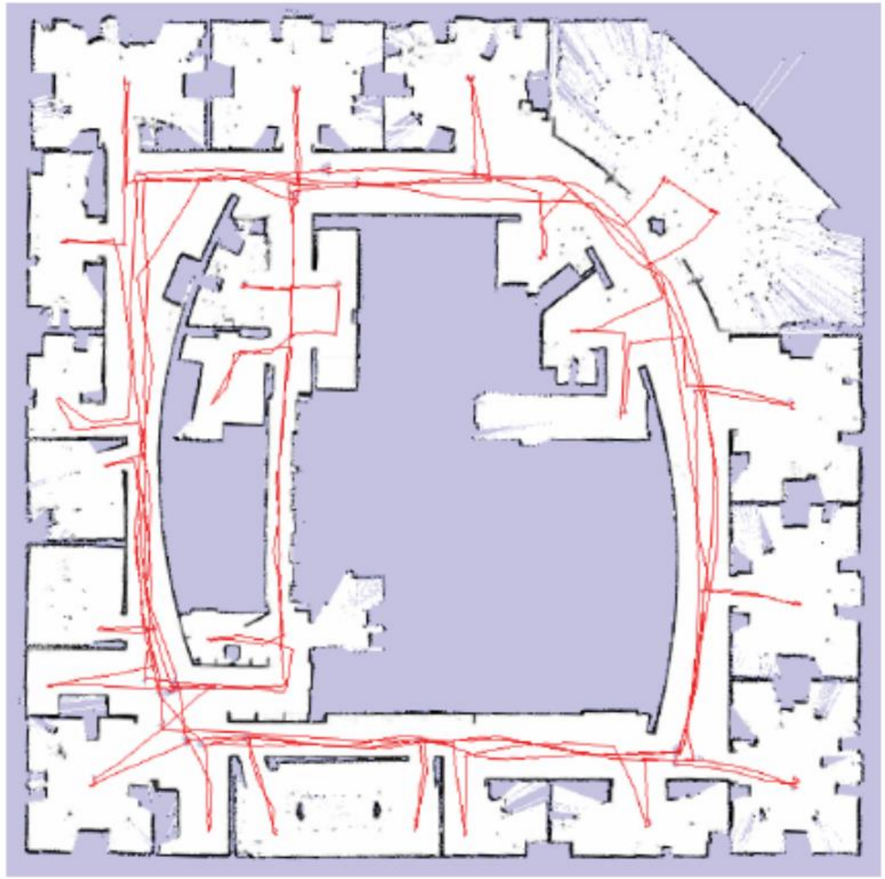
- # SLAM and Factor Graphs



What is SLAM?

Simultaneous Localization and Mapping

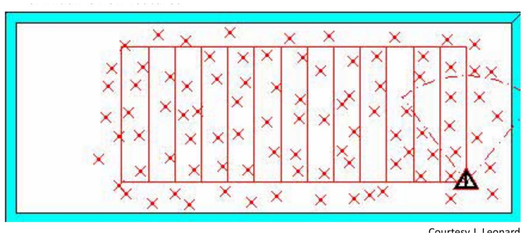
- Localization: estimate the robot pose, given a map
- Mapping: build a map, given the robot poses
- SLAM: estimate both at the same time
  - A chicken-and-egg problem
- Needed when no prior map or GPS is available
  - Indoors, underground, search-and-rescue, exploration



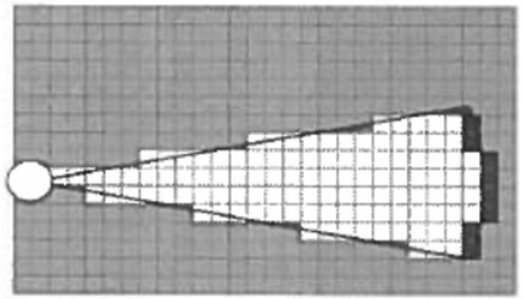
What Do We Estimate?

Robot state + a map of the environment

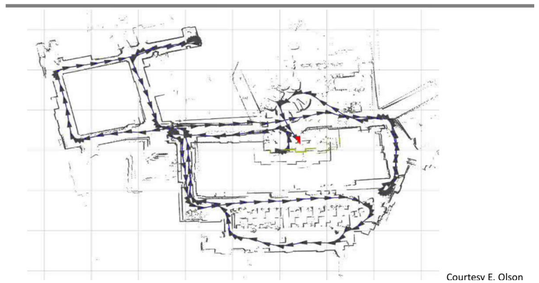
- State
  - Position, orientation, velocity
  - IMU biases, sensor calibration
  - Current pose, or the full trajectory
- Map representation
  - Landmarks, occupancy grid, poses + scans, point cloud, meshes…
  - This talk: poses + radar point cloud



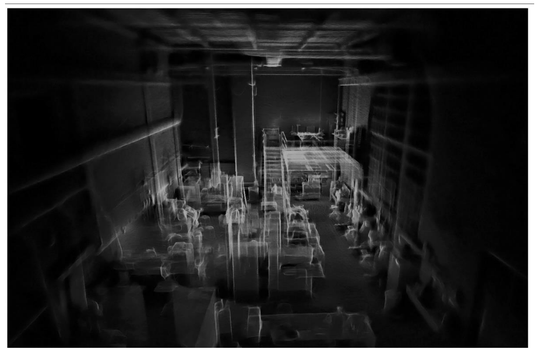
Landmarks / features



Occupancy grid



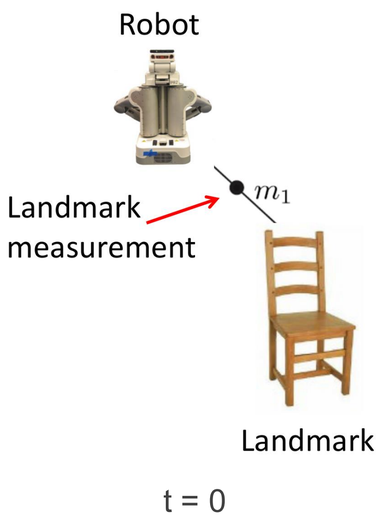
Poses + scans

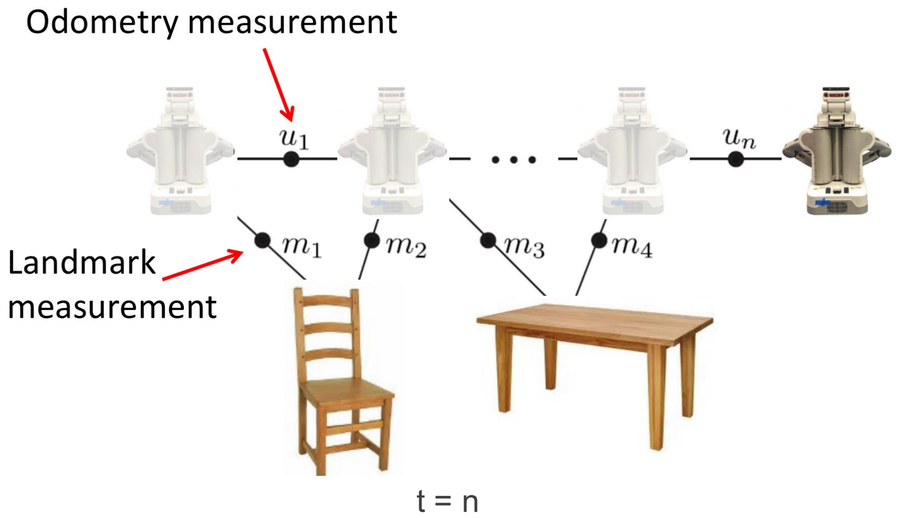


Point cloud

The SLAM Problem

A robot moves, senses, and collects noisy measurements

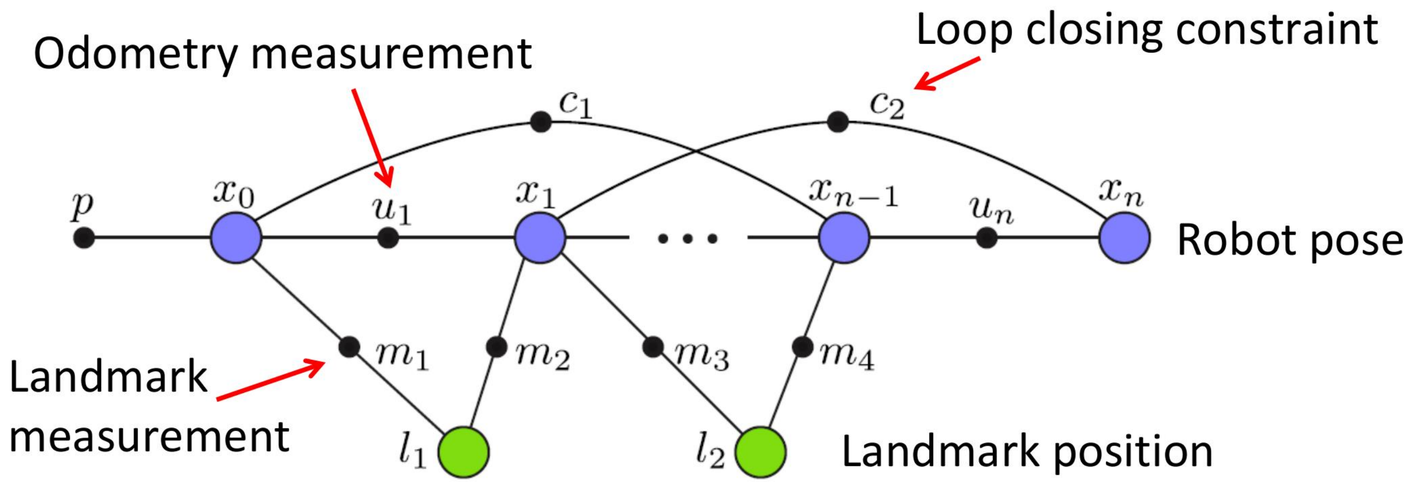




- Odometry measurements *u* (wheels, IMU, radar velocity) relate consecutive poses
- Landmark measurements *m* relate a robot pose to a landmark
- Every measurement is noisy, so errors accumulate over time (drift)

Factor Graph Representation

A bipartite graph of variable nodes and factor nodes

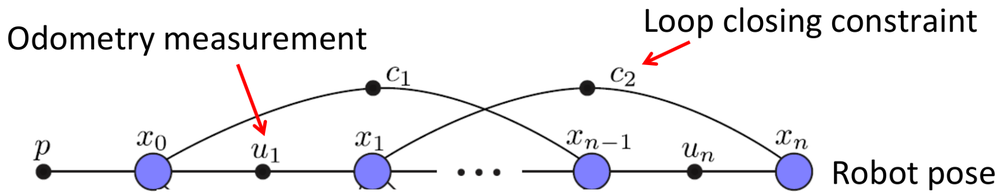


- Variable nodes (circles): the unknowns, robot poses *x* and landmark positions *l*
- Factor nodes (dots): the measurements, prior *p*, odometry *u*, landmark *m*, loop closure *c*
- Each factor constrains only the variables it touches, so the graph is sparse

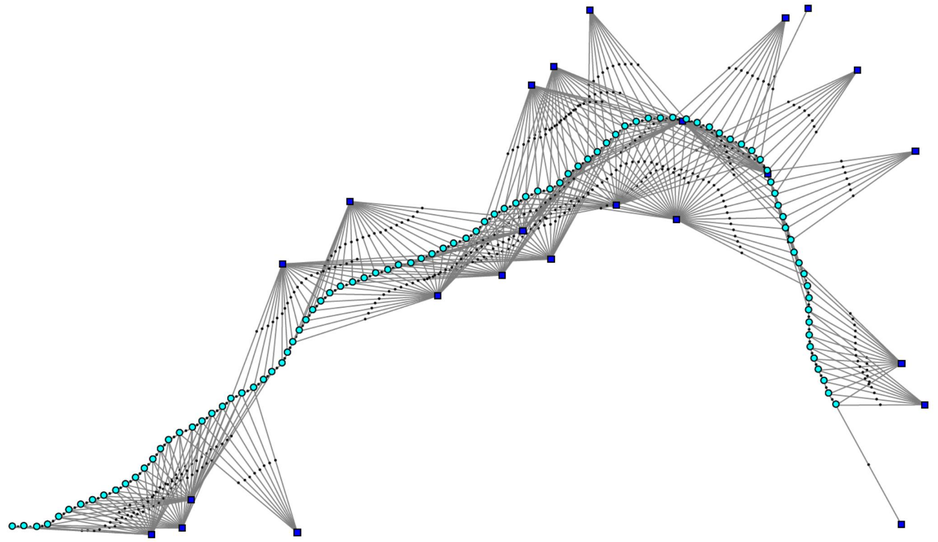
Pose Graph and Loop Closure

Constraints only between poses, no explicit landmarks

- Common for LiDAR and radar SLAM: scan matching gives relative-pose factors
- Loop closure: recognizing a revisited place adds a constraint between distant poses
  - Corrects drift accumulated by odometry
- Factors can model any sensor: IMU preintegration, radar velocity, calibration…



"Pose graph" (no explicit modeling of landmarks)



A larger factor graph SLAM example

From Factor Graph to Least Squares

Maximum a posteriori (MAP) inference

- Goal: most likely Θ given all measurements Z

$$\Theta^* = \arg\max_\Theta \, p(\Theta \mid Z) \;\propto\; p(Z \mid \Theta)\, p(\Theta)$$

- Independent measurements: one term per factor

$$p(Z \mid \Theta) = \prod_i p(z_i \mid \Theta)$$

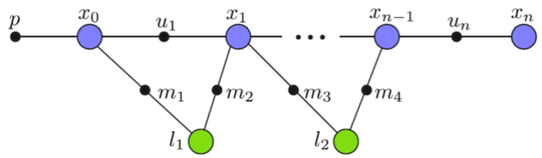
- Gaussian noise: each factor is an exponential

$$\Theta^* = \arg\max_\Theta \prod_i \exp\left(-\tfrac{1}{2}\|h_i(\Theta) - z_i\|^2_{\Sigma_i}\right)$$

- Taking −log turns the product into nonlinear least squares

$$\Theta^* = \arg\min_\Theta \sum_i \|h_i(\Theta) - z_i\|^2_{\Sigma_i}$$

- Solve iteratively (Gauss-Newton, Levenberg-Marquardt); sparsity keeps it fast (GTSAM, iSAM2)



$$\text{argmax}_\Theta \prod_{z \in Z} p\,(z|\Theta)$$

Gaussian noise

$$\text{argmin}_\Theta \sum_i \|h_i(\Theta) - z_i\|^2$$

$h()$ linear

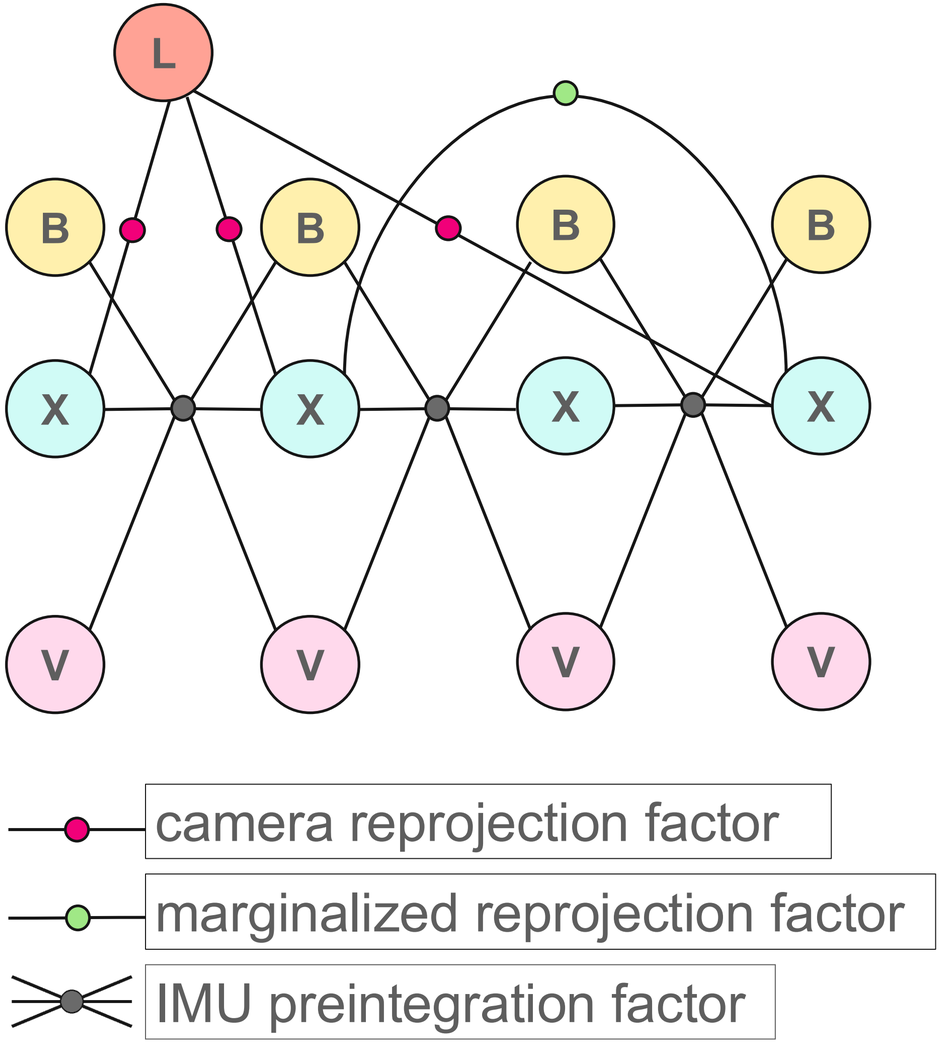
$$\text{argmin}_x \|Ax - b\|^2$$

Factor Graph

Visual-inertial odometry

Factor Graph of different Odometry Method

- Visual Inertial Odometry
  - Constrain on pose relation or relation between pose and landmarks
- LiDAR Inertial Odometry
  - Constrain on pose or between pose

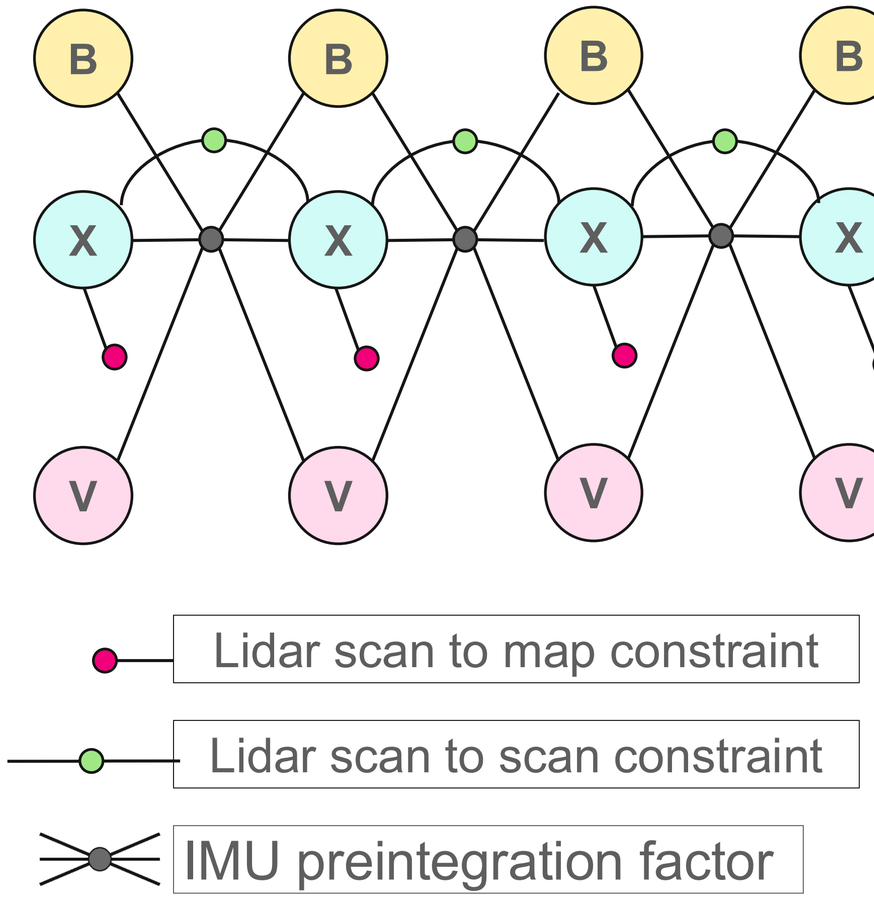


Factor Graph

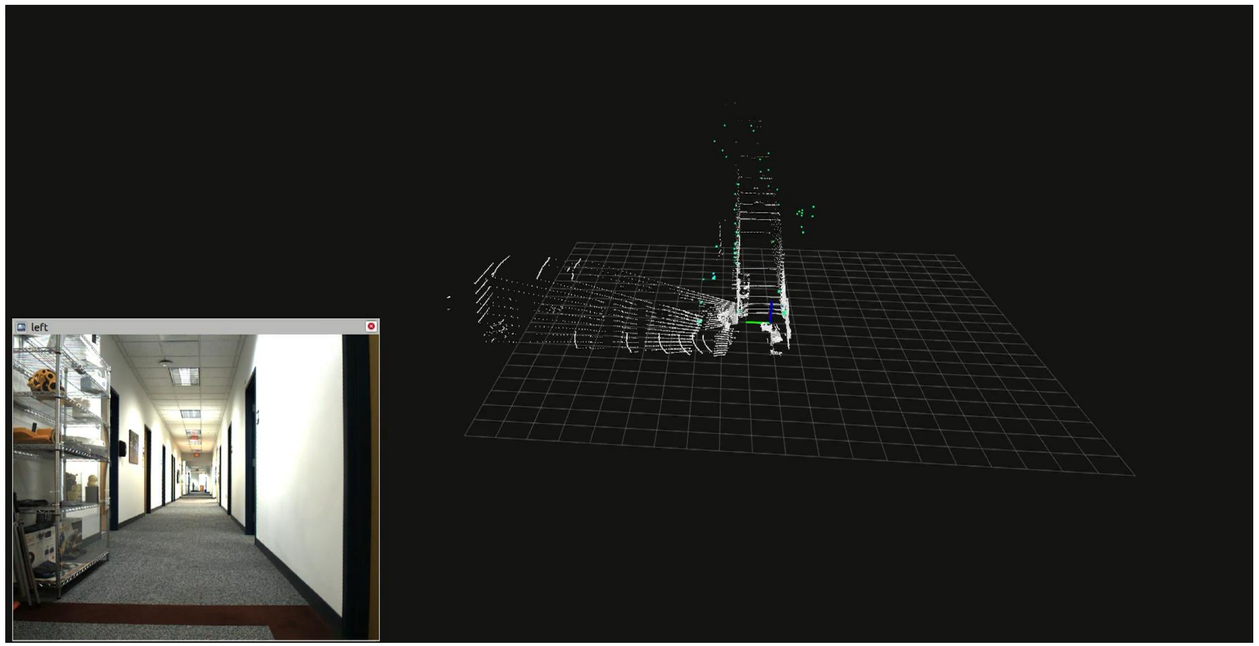
LiDAR-inertial odometry

Factor Graph of different Odometry Method

- Visual Inertial Odometry
  - Constrain on pose relation or relation between pose and landmarks
- LiDAR Inertial Odometry
  - Constrain on pose or between pose

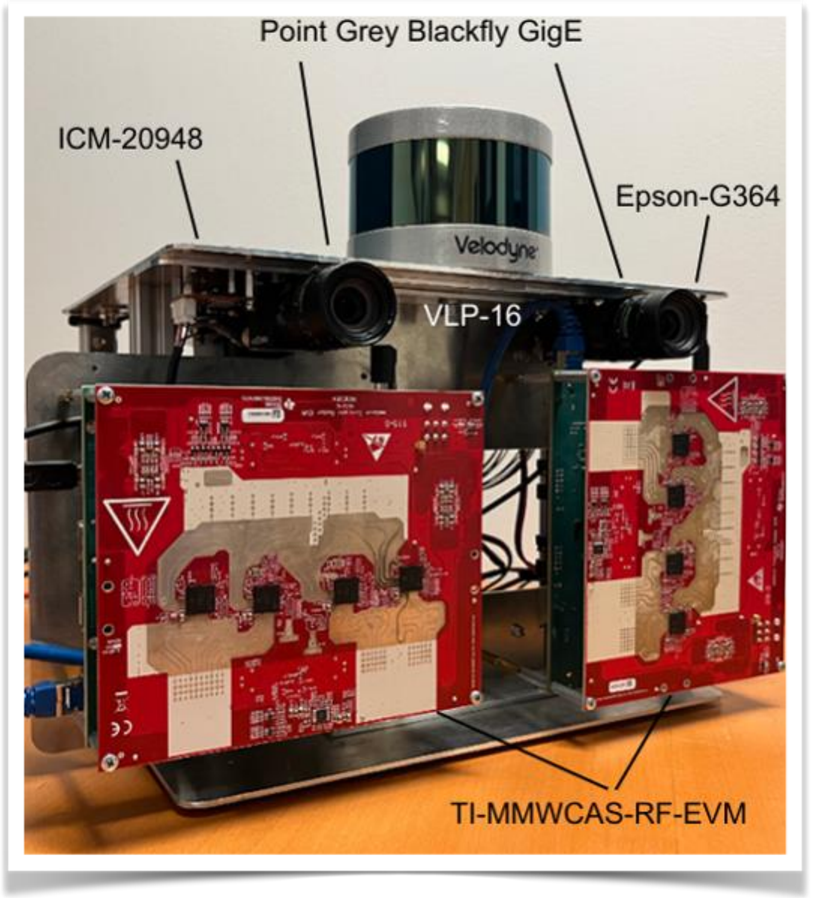


- **SLAM with Radar Point Cloud**



Radar Sensor Rig

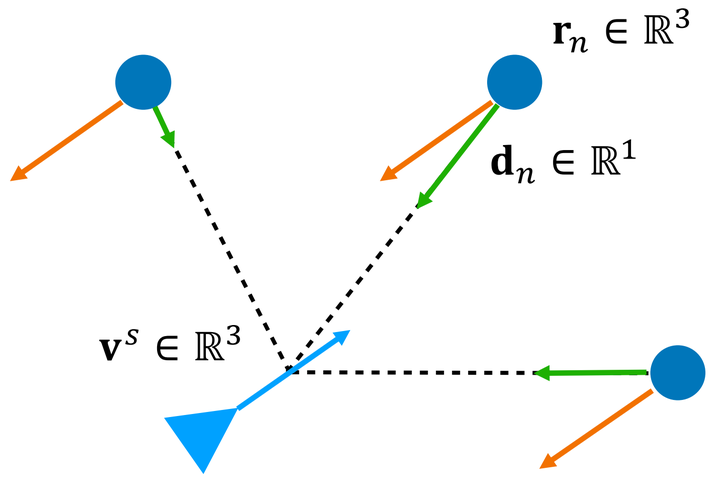
- Radars are triggered at their own schedule with separated operational frequencies
- Configurations:
  - Start frequency: 77 GHz, 79 GHz
  - Wavelength: ~ 4 mm
  - Frequency slope: 40 MHz/us
  - Bandwidth: 1.5 GHz
  - Frame rate: 10 Hz



Radar Sensor-frame Velocity

- Estimate linear sensor frame velocity from radar point clouds
  - point position : $\mathbf{r}_n \in \mathbb{R}^3$
  - Doppler velocity : $\mathbf{d}_n \in \mathbb{R}^1$
  - sensor frame velocity : $\mathbf{v}^s \in \mathbb{R}^3$

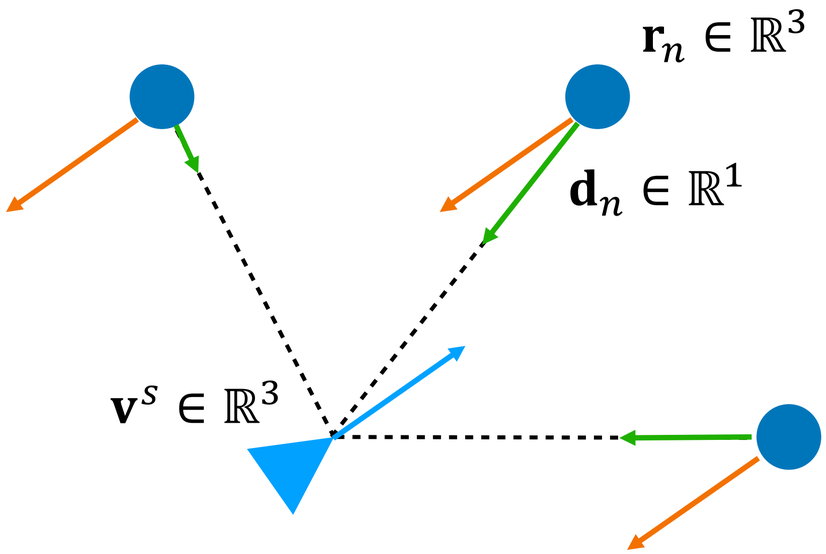
$$\mathbf{v}^s = \underset{\mathbf{v}^s}{\operatorname{argmin}} \left(\sum_n || (\mathbf{v}^s)^\top \frac{\mathbf{r}_n}{||\mathbf{r}_n||} + \mathbf{d}_n ||^2_{\Sigma_d} \right)$$



Radar Sensor-frame Velocity

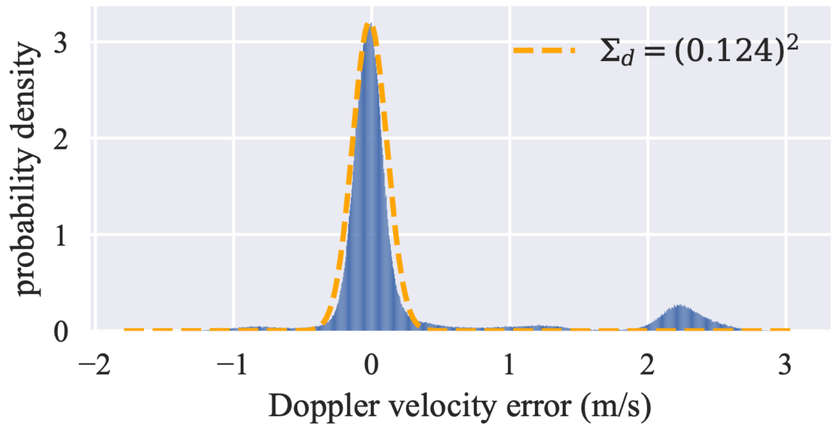
- Estimate sensor frame velocity uncertainty:
  - Doppler velocity variance : $\Sigma_d \in \mathbb{R}^1$
  - Point direction : $A_n = \frac{\mathbf{r}_n}{||\mathbf{r}_n||}$
  - Velocity covariance : $\Sigma_{\mathbf{v}^s} \in \mathbb{R}^{3\times3}$

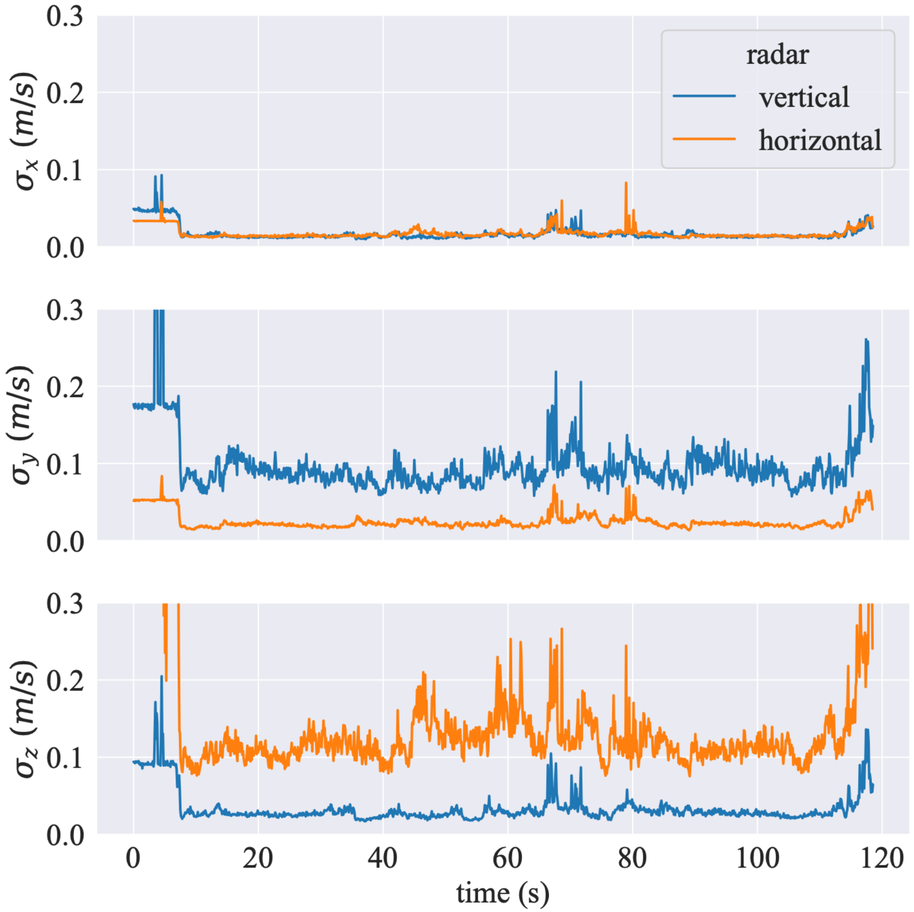
$$\Sigma_{\mathbf{v}^s} = \left(A^\top \Sigma_d^{-1} A\right)^{-1}$$



Radar Sensor-frame Velocity

- Evaluate Doppler velocity uncertainty with VIO pseudo ground truth
- Dual radar layout to compensate measurement discrepancy





Radar Sensor-frame Velocity

- **To remove outliers: Cauchy robust loss**
  - Normalized Doppler residual of point n

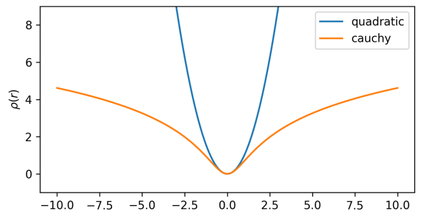
$$s_n = \frac{1}{\sigma_d^2}\left(d_n - \mathbf{v}^\top \frac{\mathbf{r}_n}{\|\mathbf{r}_n\|}\right)^2$$

  - Cauchy loss grows only logarithmically

$$\rho_{\text{quad}}(s) = s \qquad \rho_{\text{Cauchy}}(s) = \log(1+s)$$

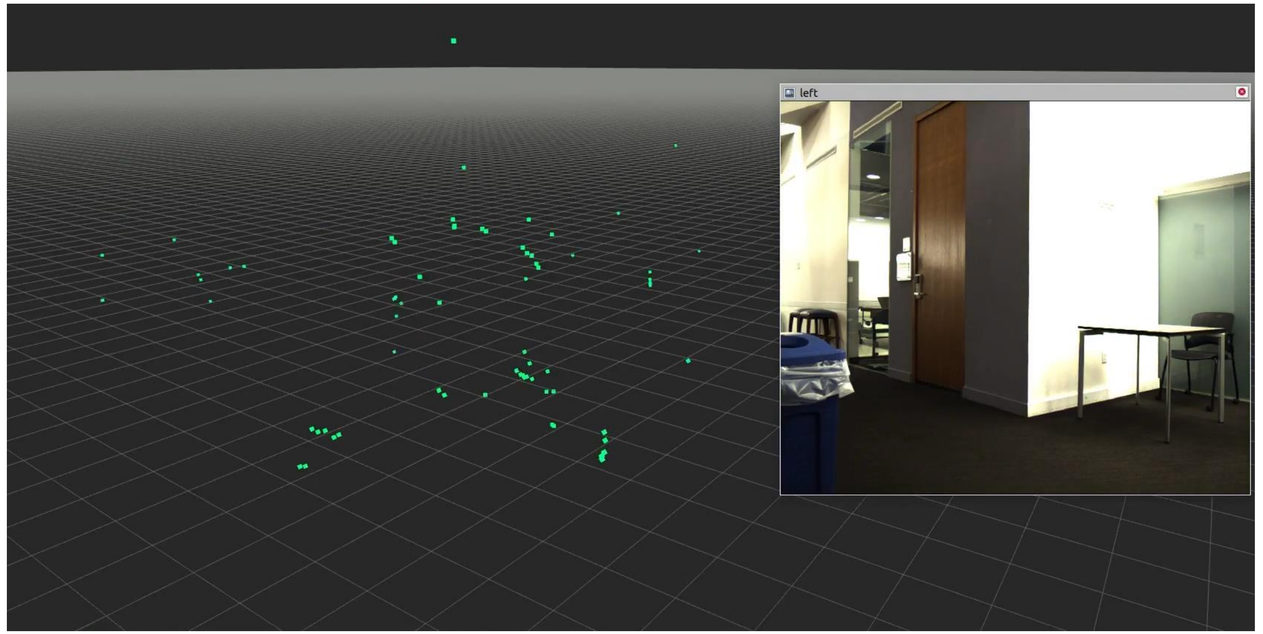
  - Equivalent to reweighting each point

$$\mathbf{v}^* = \arg\min_{\mathbf{v}} \sum_n \rho(s_n), \qquad w_n = \rho'(s_n) = \frac{1}{1+s_n}$$



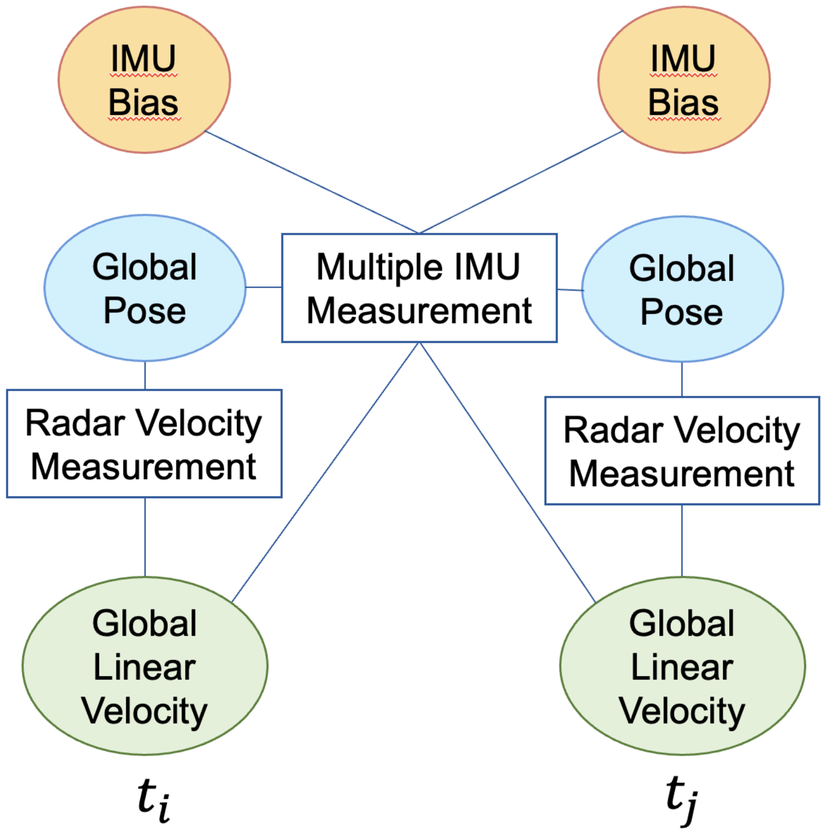
  - Outliers (moving objects, multipath) get a small weight instead of dragging v
  - Initial value from IMU-integrated linear velocity

2X



Linear body frame velocity (red bar)

Radar Inertial Odometry



Mathematical Dependency
(Factor Graph)

- IMU pre-integration factor:

  integrate linear acceleration, rotation velocity,

  model slow varying IMU bias

$$\mathbf{r}_{\Delta \mathbf{R}_{ij}} = Log\left(\Delta \tilde{\mathbf{R}}_{ij}^{\top}\left(\mathbf{R}_i^{\top}\mathbf{R}_j\right)\right) \tag{10}$$

$$\mathbf{r}_{\Delta \mathbf{v}_{ij}} = \mathbf{R}_i^{\top}\left(\mathbf{v}_j - \left(\mathbf{v}_i + \mathbf{g}\Delta t_{ij}\right)\right) - \Delta \tilde{\mathbf{v}}_{ij} \tag{11}$$

$$\mathbf{r}_{\Delta \mathbf{p}_{ij}} = \mathbf{R}_i^{\top}\left(\mathbf{p}_j - \left(\mathbf{p}_i + \mathbf{v}\Delta t_{ij} + \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right)\right) - \Delta \tilde{\mathbf{p}}_{ij} \tag{12}$$

$$\mathbf{r}_{\Delta \mathbf{b}_{ij}^a} = \mathbf{b}_j^a - \mathbf{b}_i^a \tag{13}$$

$$\mathbf{r}_{\Delta \mathbf{b}_{ij}^g} = \mathbf{b}_j^g - \mathbf{b}_i^g \tag{14}$$

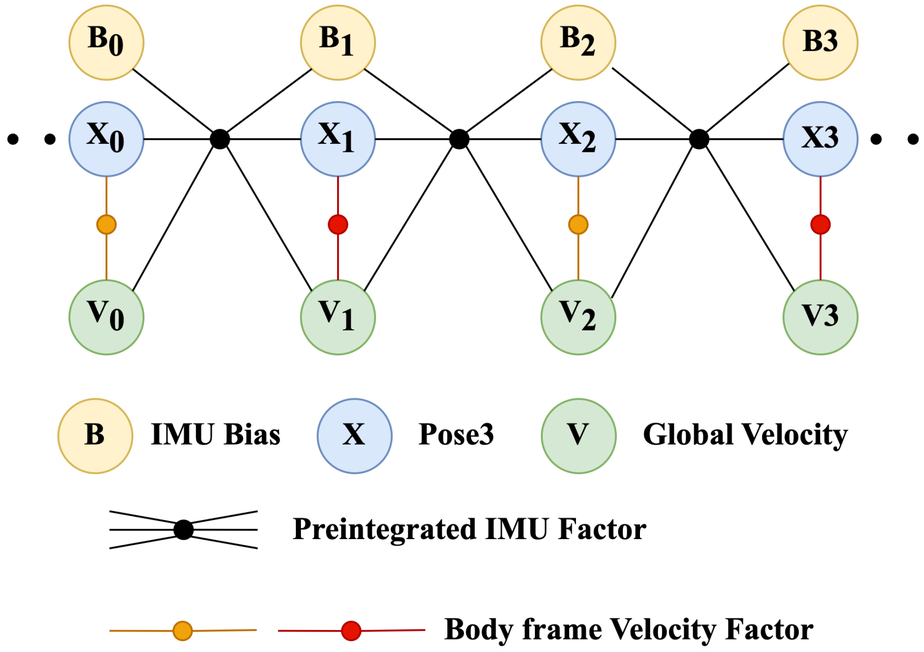
- Radar velocity factor:

  constraint integrated velocity, integration direction

$$\mathbf{r}_{\mathbf{V}_j^b} = \mathbf{R}_j^{\top}\mathbf{v}_j - \mathbf{v}_j^b \tag{16}$$

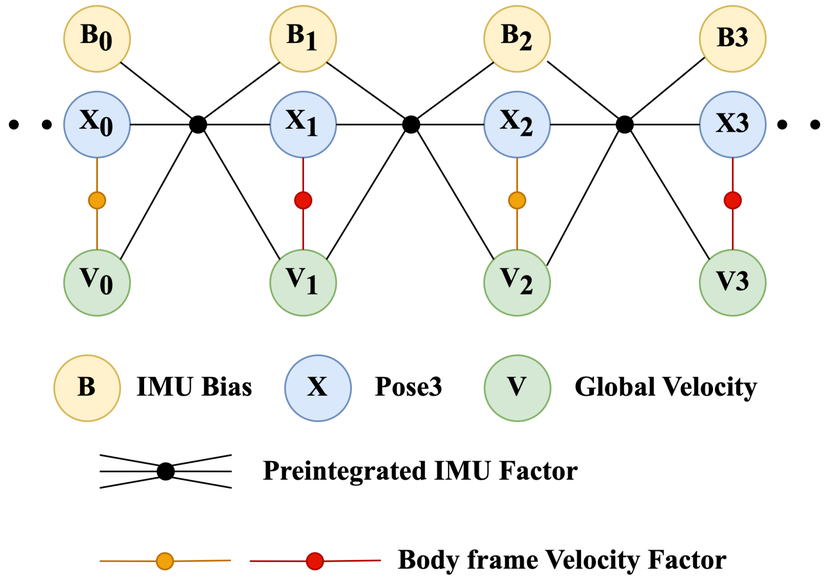
Multi-Radar Inertial Odometry

- Using body frame velocity measured from **radar** to correct IMU velocity integration and integration direction.
- **Body frame velocity factor:**
  - Residual : $\mathbf{r}_{\mathbf{v}_j^b} = \mathbf{R}_j^\top \mathbf{v}_j - \mathbf{v}_j^b$
  - Jacobian w.r.t. rotation : $\mathbf{J}_R = [\mathbf{R}^\top \mathbf{v}]_\times$
  - Jacobian w.r.t. velocity : $\mathbf{J}_\mathbf{v} = \mathbf{R}^\top$

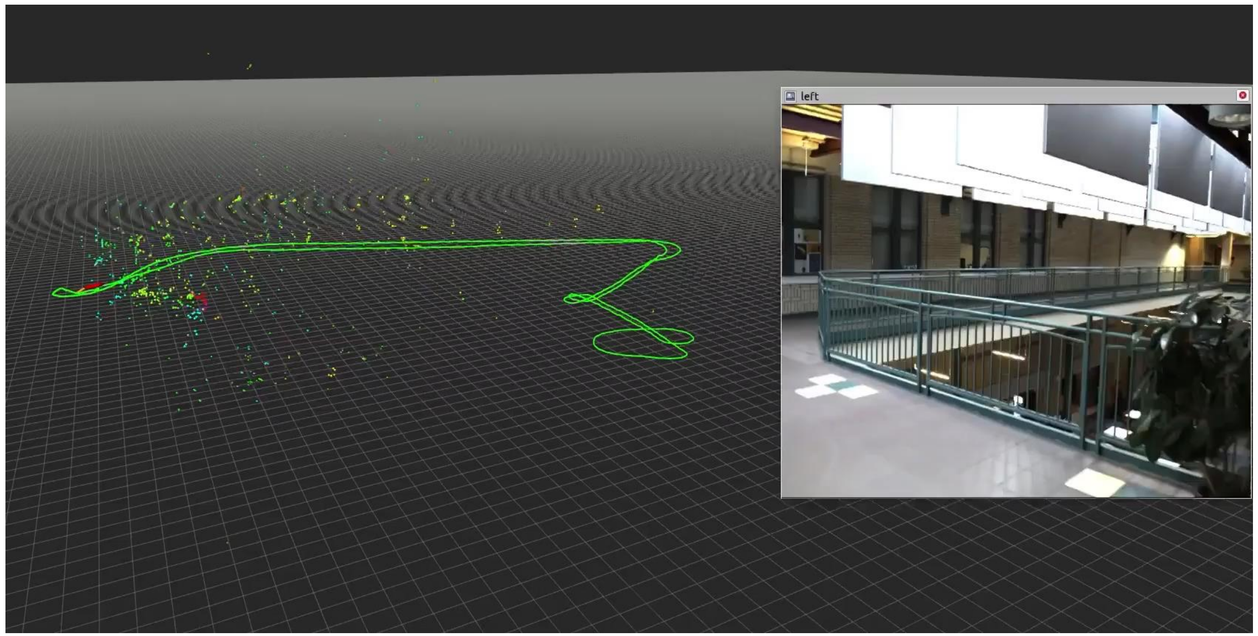


Multi-Radar Inertial Odometry

- Integrates body frame velocity with IMU measurements using **fixed-lag smoothing**:
  - 5 seconds optimization window
  - IMU static initialization



4X



VIO pseudo ground truth (green trajectory)

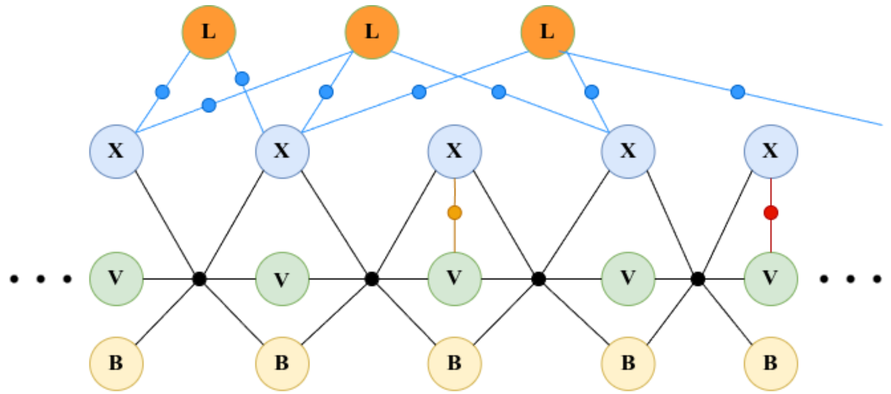
Radar Map

4X



Radar-Visual-Inertial Odometry

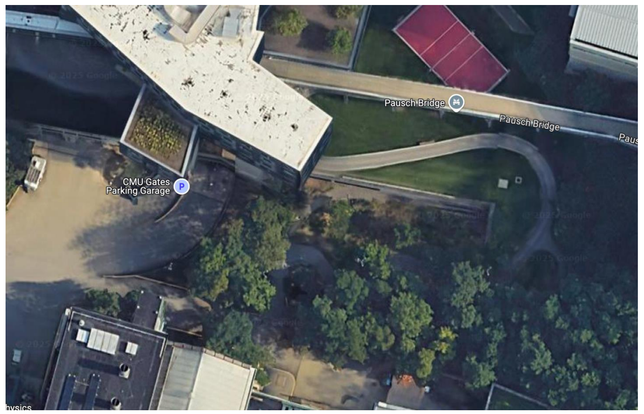
- **Stereo Visual frontend**
  - optical flow tracker
- **Radar frontend**
  - velocity optimization
- **Backend**
  - IMU preintegration factor
  - Stereo reprojection factor
  - Body frame velocity factor

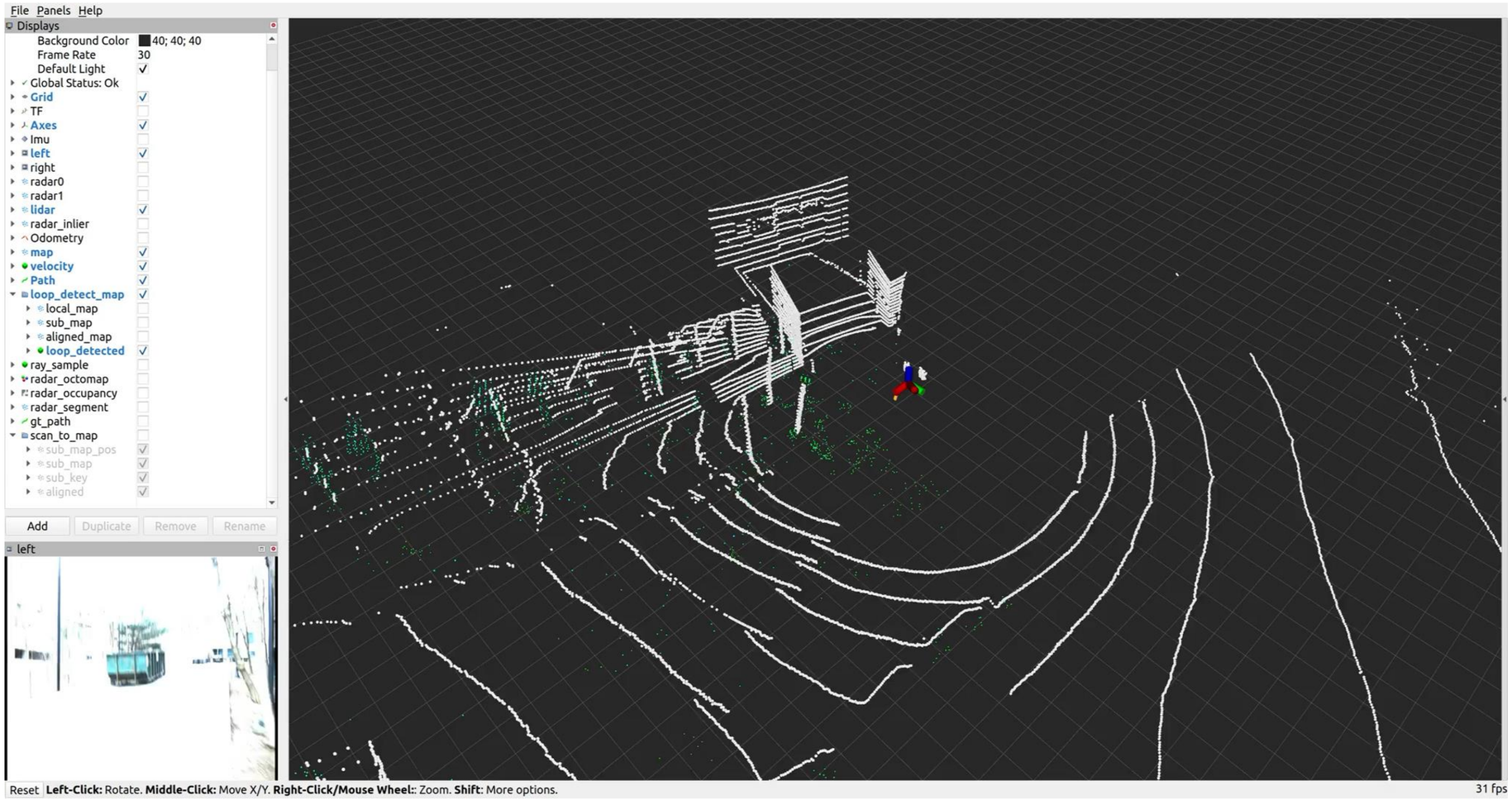




Outdoor Radar-Inertial Odometry

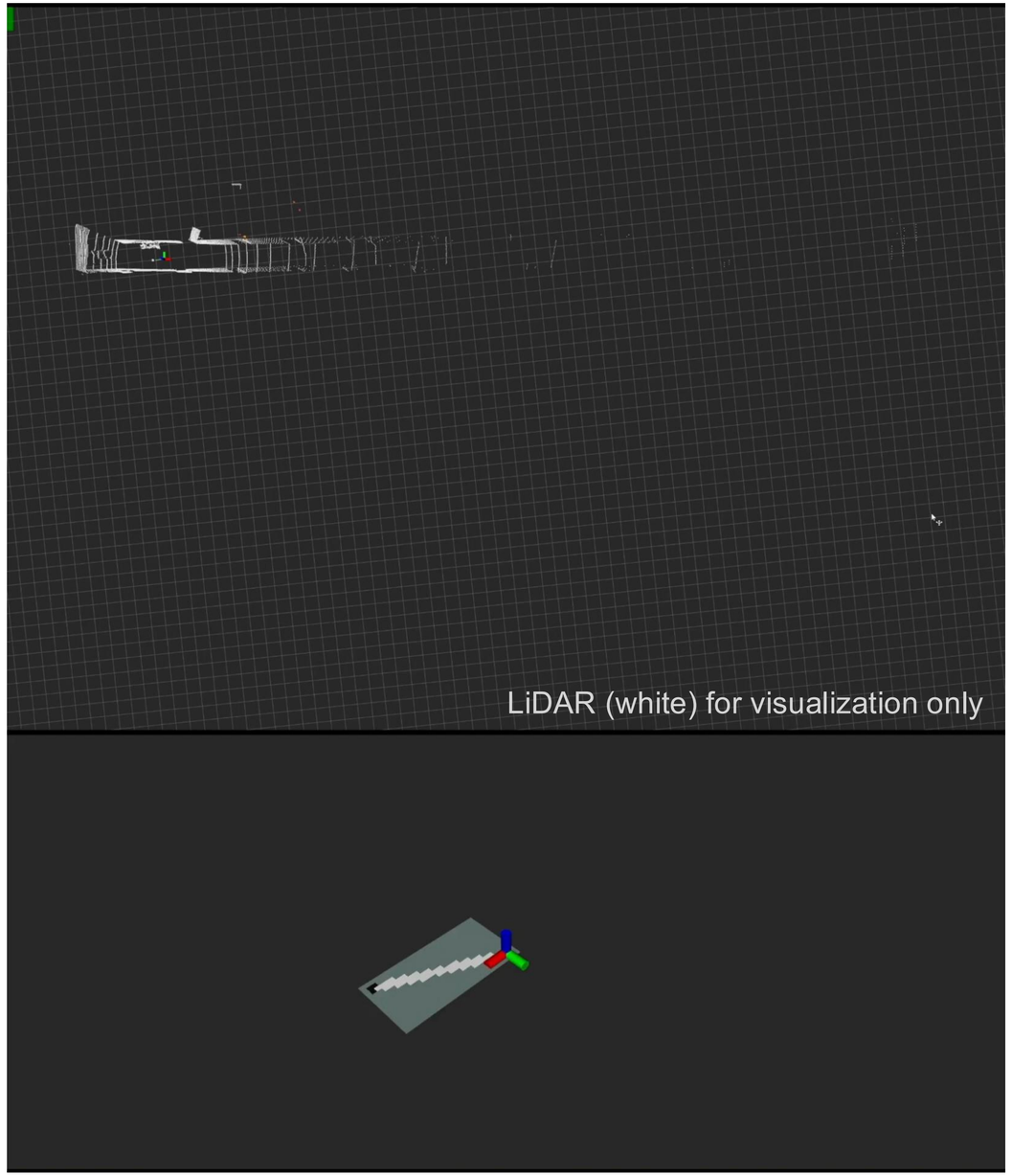
- **Campus outdoor garden uphill**
- **White LiDAR points for visualization only**
- **Reference :**





Mapping & Loop Close

- Loop close for drift
  - Feature based global registration
- 3D Occupancy Mapping using OctoMap
- Projected 2D Occupancy Map



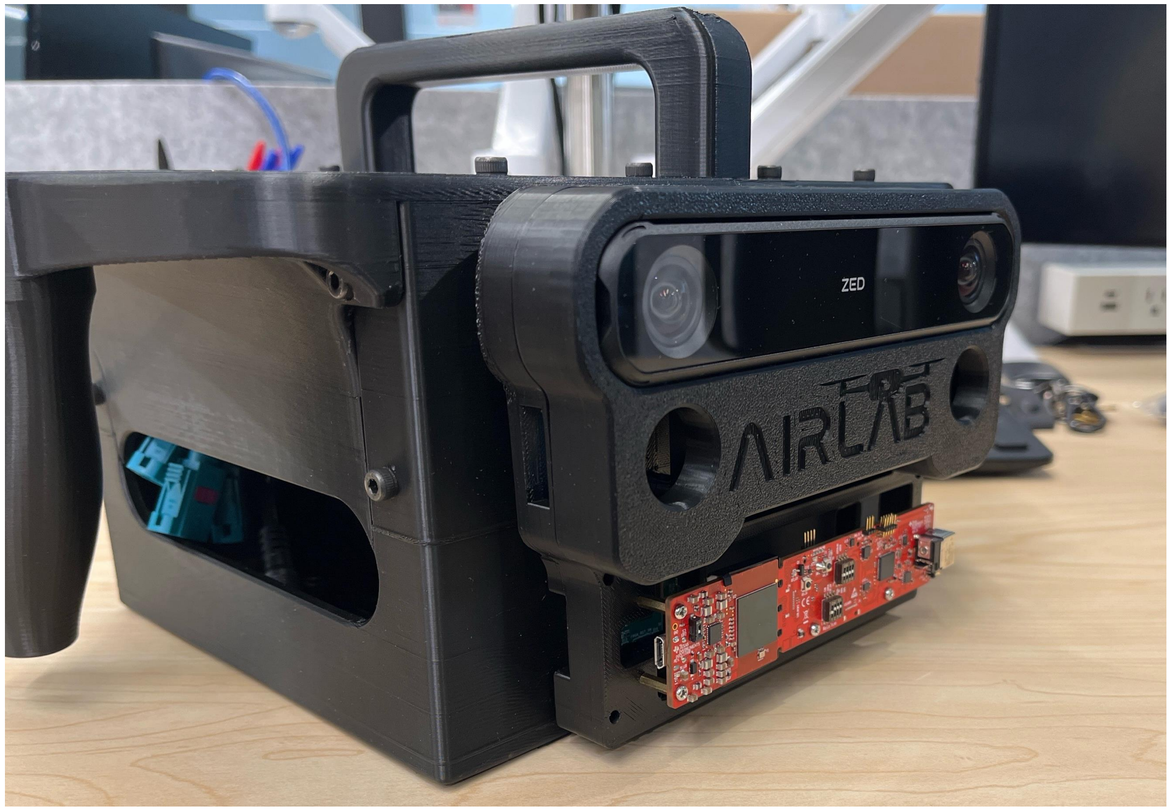
Localization

- Adaptive Monte Carlo Localization
  - Particle Filter
  - Motion Model
    - Radar Inertial Odometry
  - Sensor Model
    - Accumulate and project radar points to 2D scan
  - Given Initial position

4X

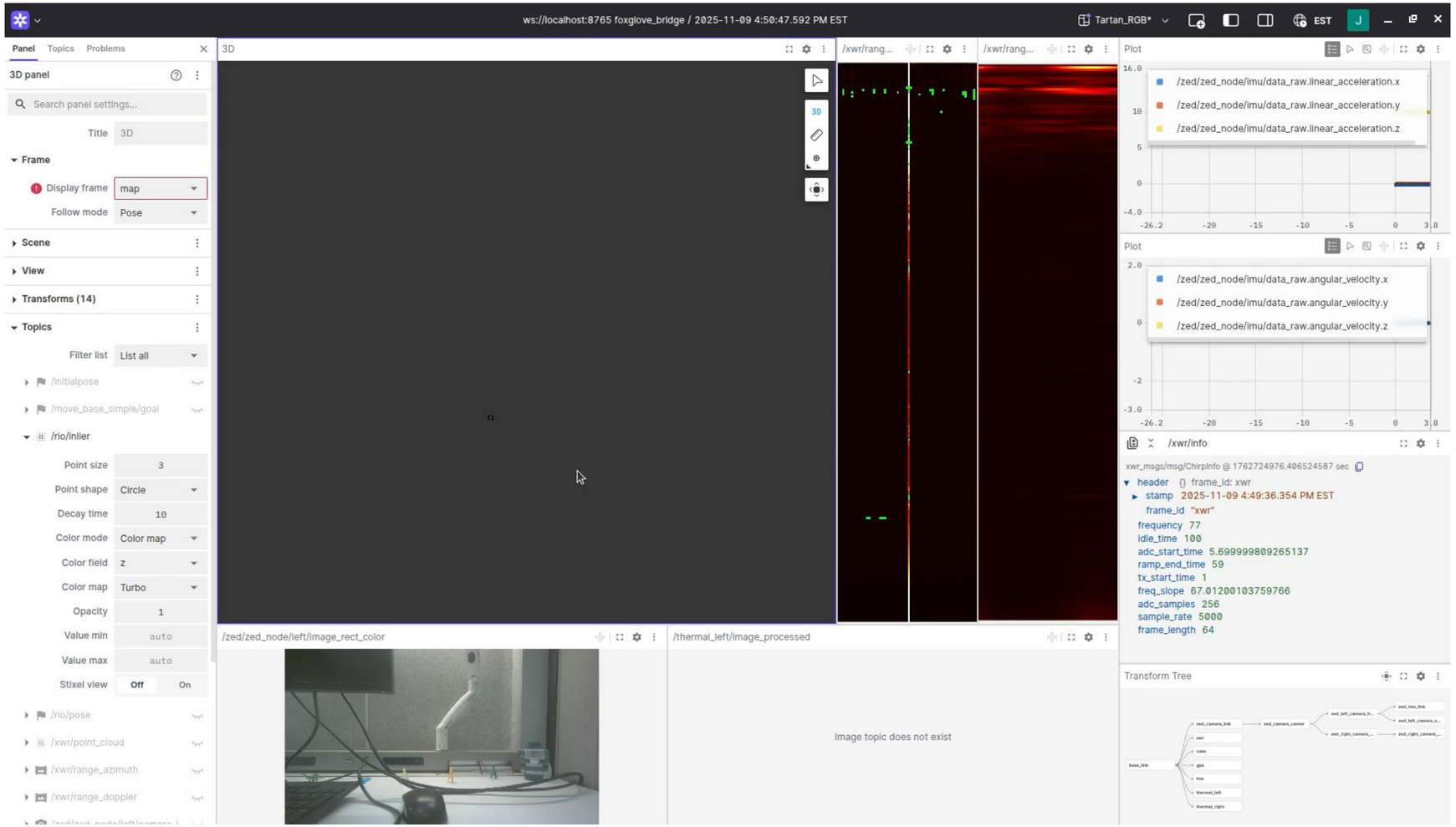


- **Radar Inertial Odometry with TI AOP Radar**



AOP Radar

Realtime signal processing & RIO



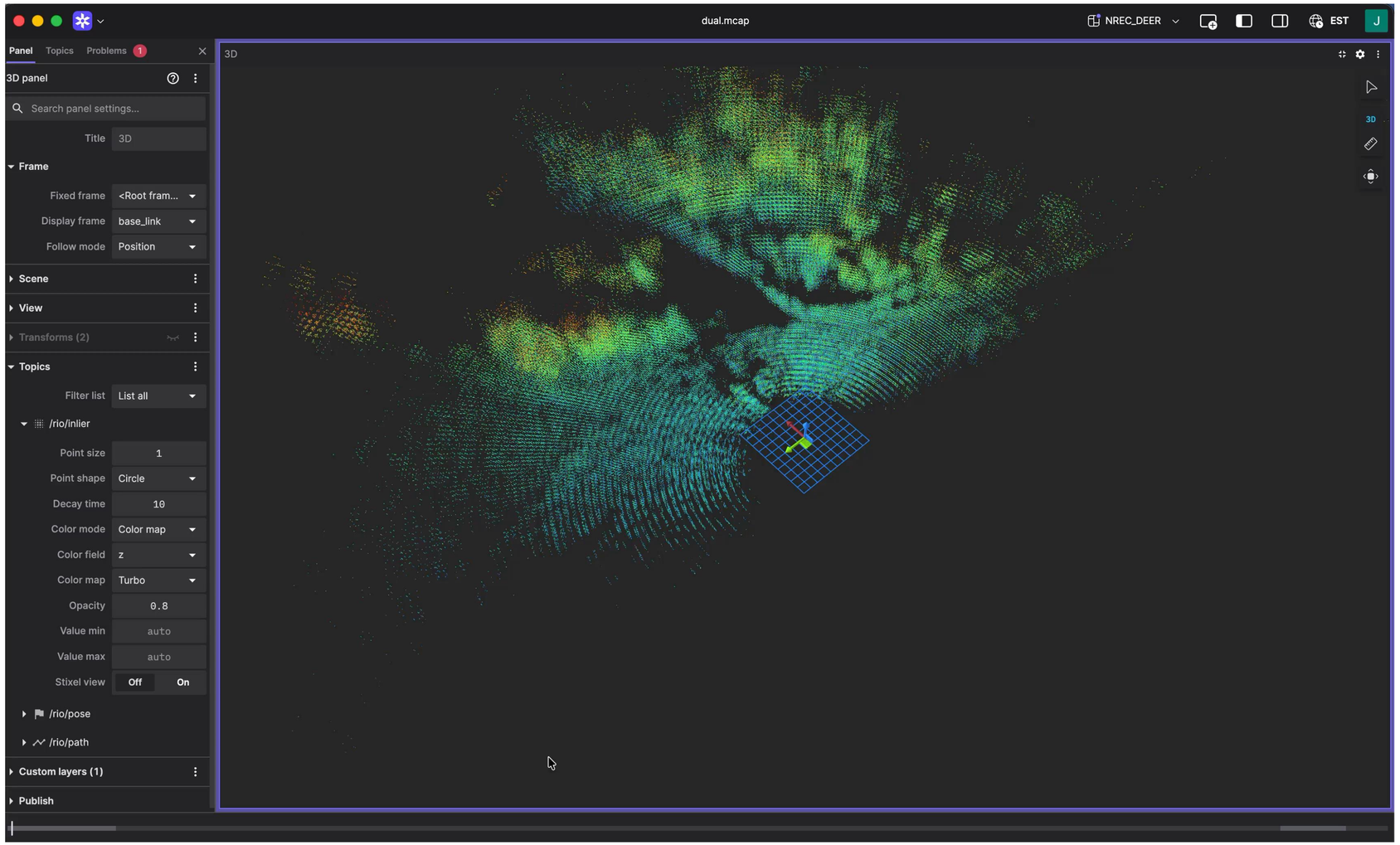
Radar Inertial Odometry with Commercial Radar

Sensrad Hugin Radar

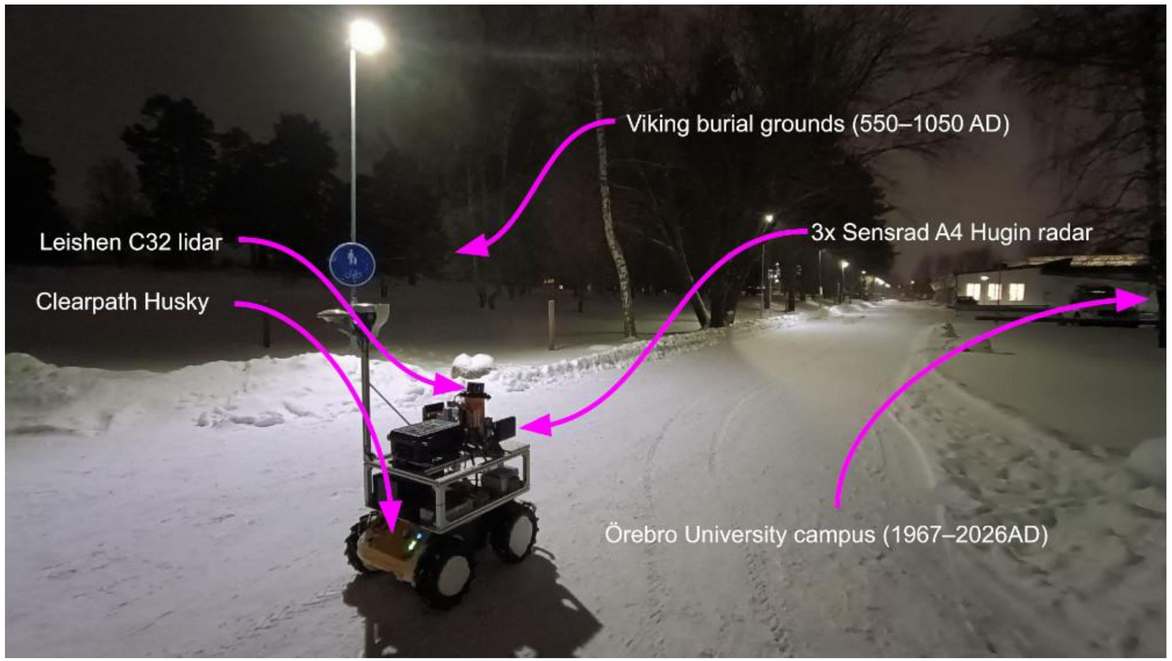


Commercial Radar

Dual Radar Inertial Odometry



- # ICRA 2026 Radar SLAM Challenge

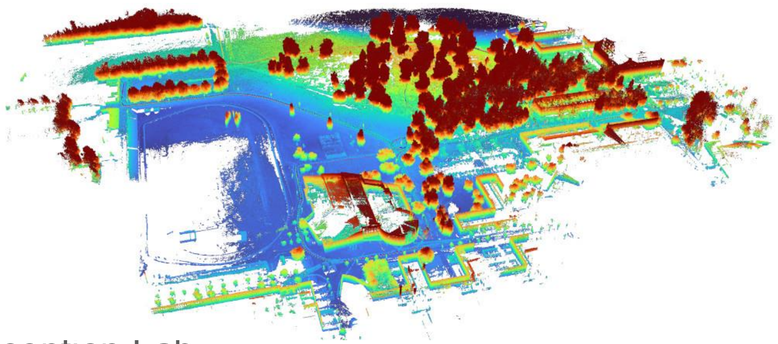


Challenge Dataset

Triple Hugin Radar

Two logs

- **01_campus_training_localized** (2257s)
  1.8km
- **02_campus_eval_filtered** (3813s)
  2.6km

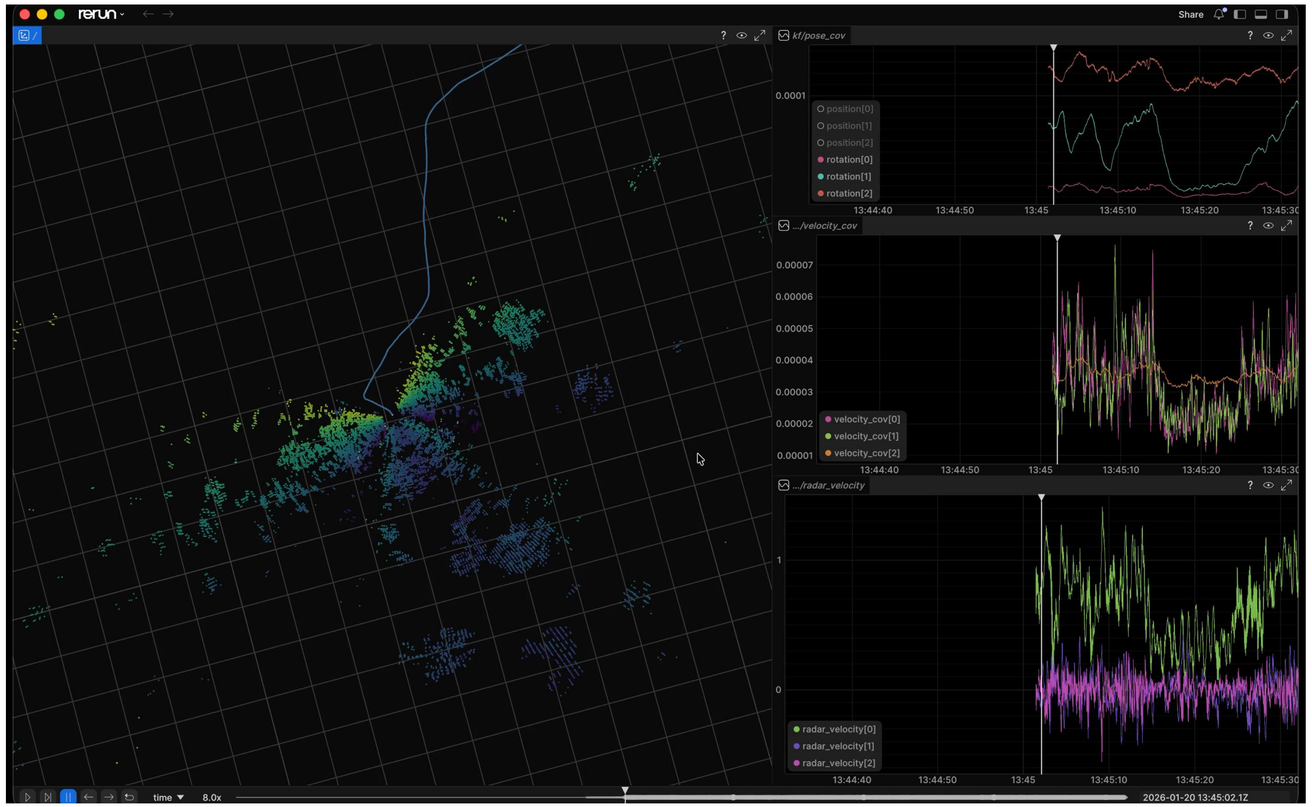




Multi-Radar Inertial Odometry

ICRA 2026 Radar SLAM Challenge

- Use all three radars + IMU pre-integration [1]
- No scan matching
- Build a body-velocity factor when a new frame arrives
- Hard constraint on marginalized pose priors keeps uncertainty from growing unboundedly



[1] Huang, Jui-Te, et al. "Multi-radar inertial odometry for 3D state estimation using mmWave imaging radar." ICRA 2024

Submaps & Global Descriptors

ICRA 2026 Radar SLAM Challenge

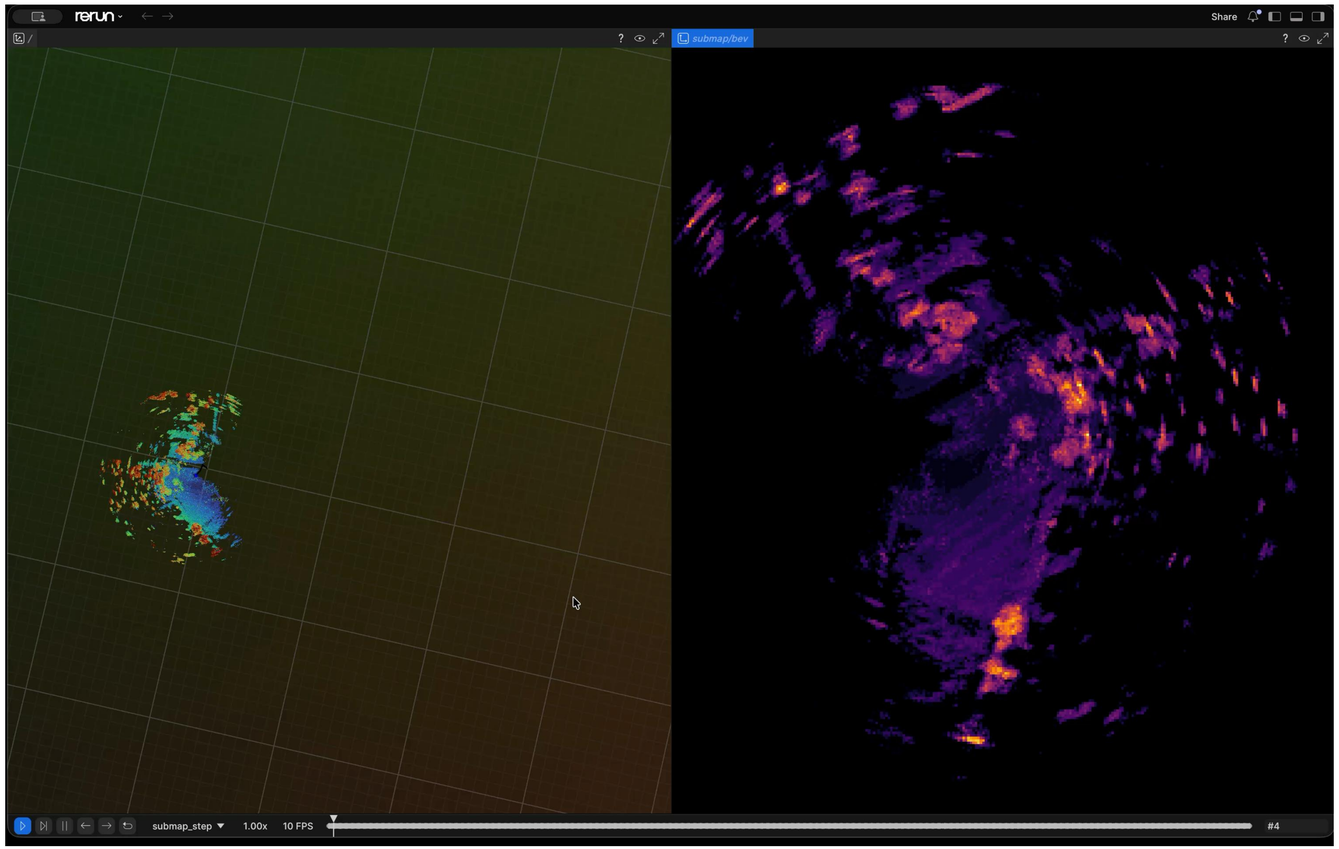
- Create a new submap every 5 meters
- Create BEV images according to point density
- Feed BEV images through foundation image models; feature layers give the global descriptor

Baselines

- BEVPlace++
- ResNet-18

Foundation models

- vit-base-patch16-224
- dinov2-base, dinov2-large
- dinov3-vitl16-pretrain-sat493m
- dinov3-vitl16-pretrain-lvd1689m



Submaps & Global Descriptors

ICRA 2026 Radar SLAM Challenge

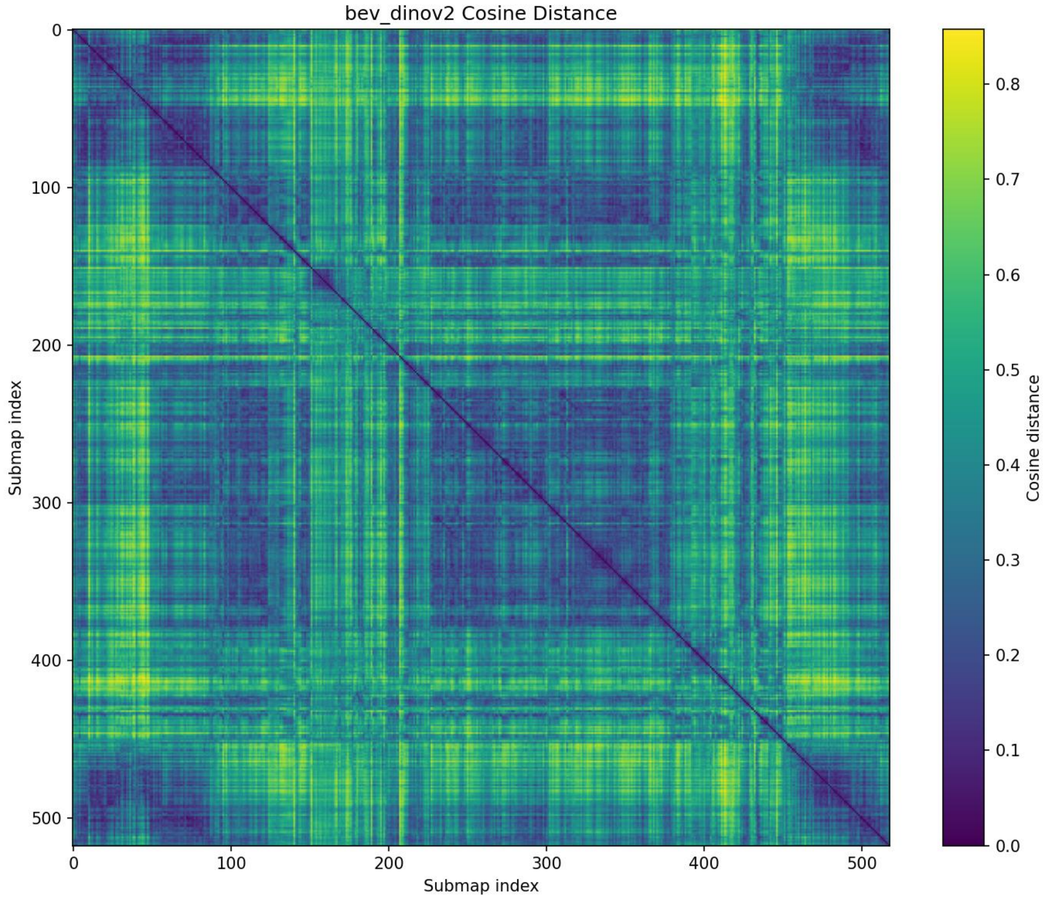
- Create a new submap every 5 meters
- Create BEV images according to point density
- Feed BEV images through foundation image models; feature layers give the global descriptor

Baselines

- BEVPlace++
- ResNet-18

Foundation models

- vit-base-patch16-224
- dinov2-base, dinov2-large
- dinov3-vitl16-pretrain-sat493m
- dinov3-vitl16-pretrain-lvd1689m

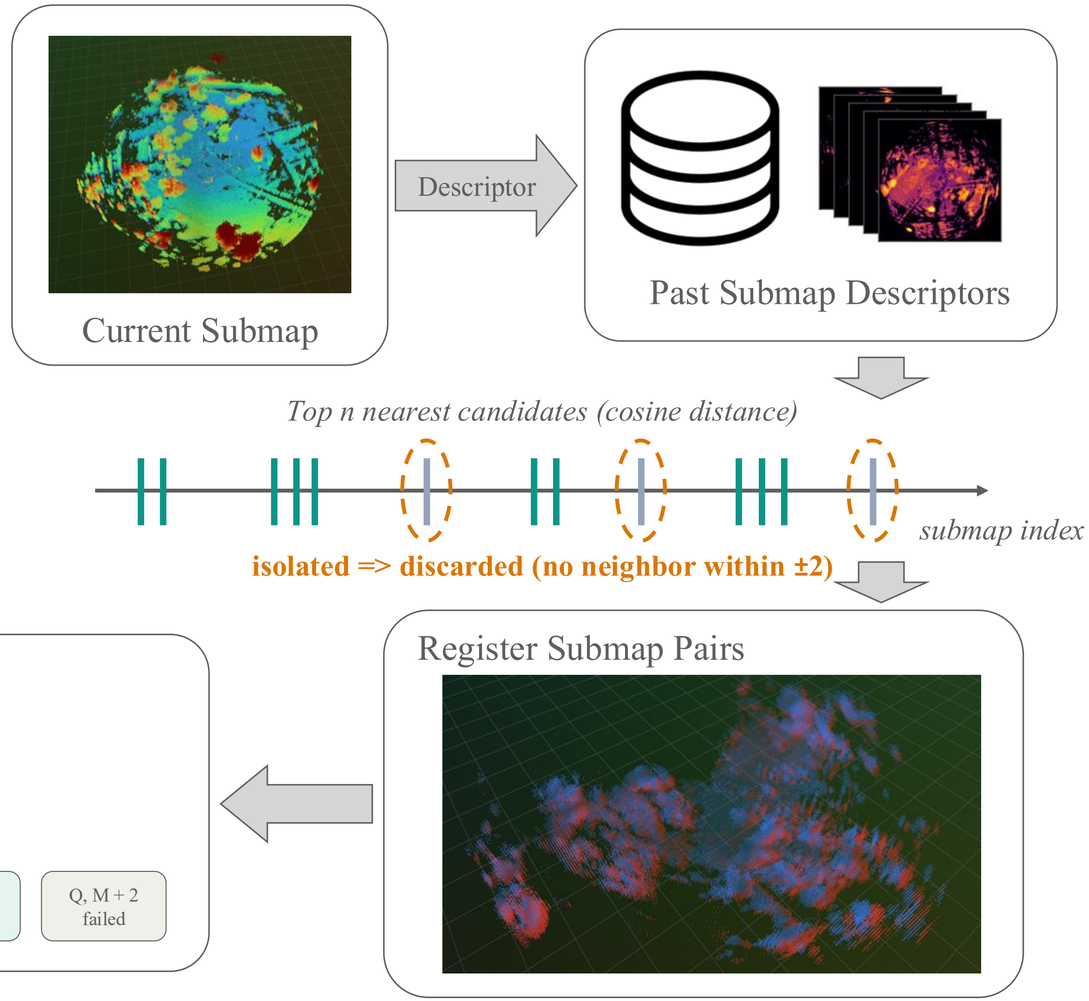


Loop Detection

ICRA 2026 Radar SLAM Challenge

- Cheap candidates from descriptors
- Filter isolated candidates
- Register submaps: KISS-Matcher
- Verify neighborhood

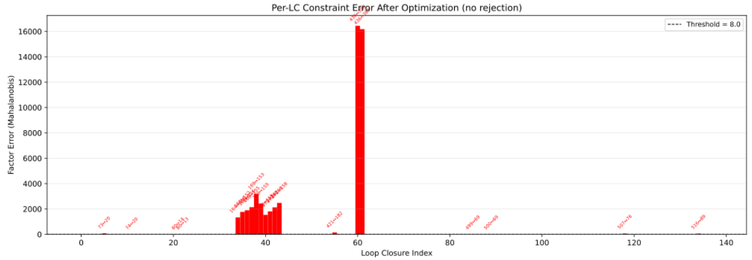
In essence: good old-fashioned engineering



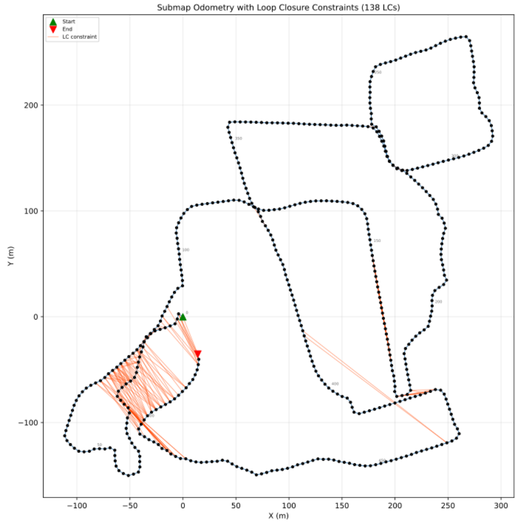
Loop Close

ICRA 2026 Radar SLAM Challenge

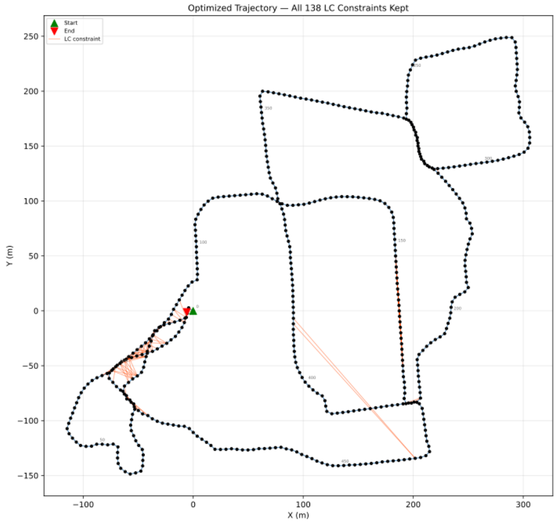
- Pose graph optimization with a Cauchy robust loss
- Two-step optimization
- Second step throws away outlier constraints



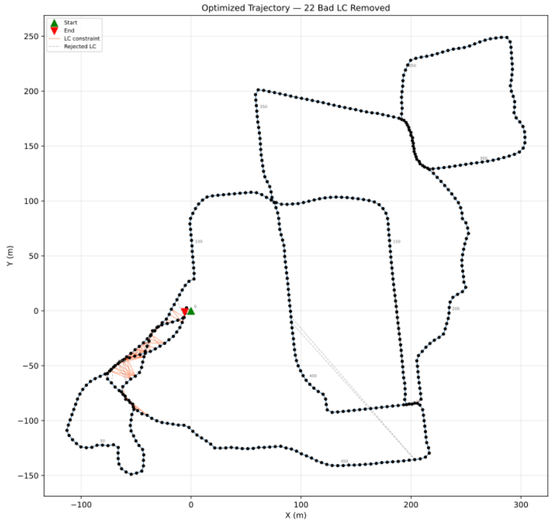
LC factor error



Before loop closed



First step



Second step

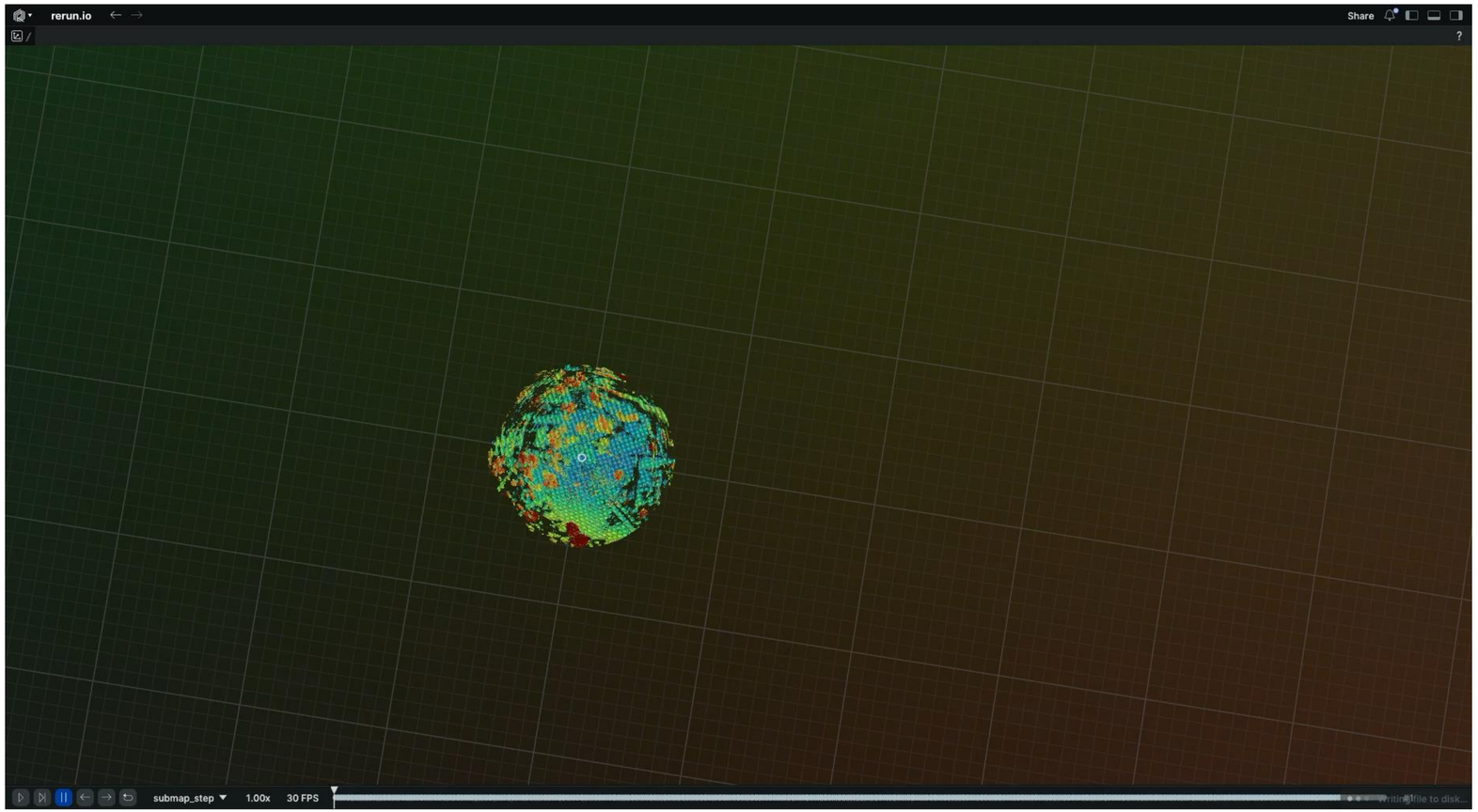
Final Results

ICRA 2026 Radar SLAM Challenge

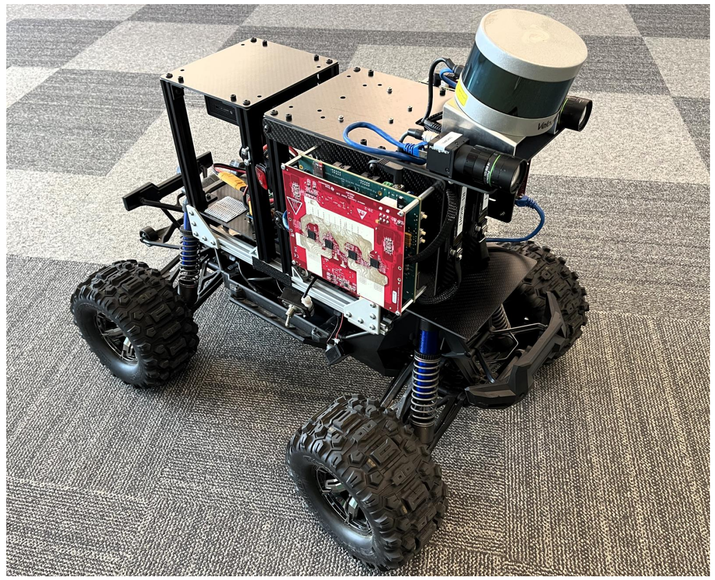
2.35 m

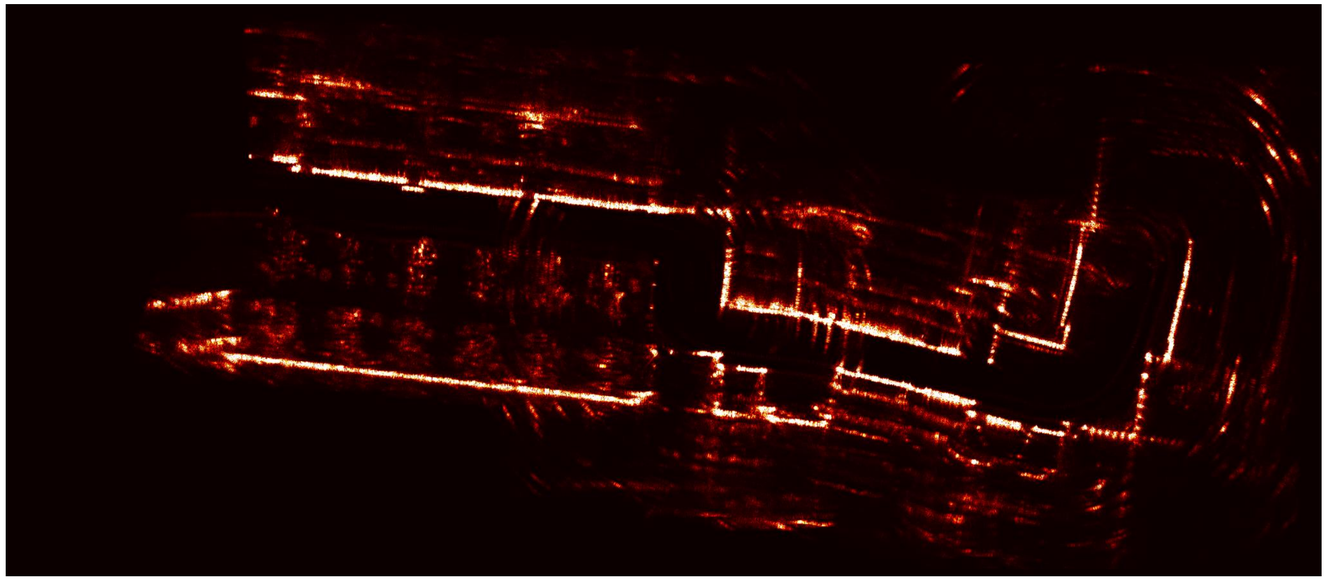
Absolute trajectory error (ATE) of the final SLAM result
Total distance: 2.6 km

1 st place !



- # Synthetic Aperture Radar For Indoor Mapping

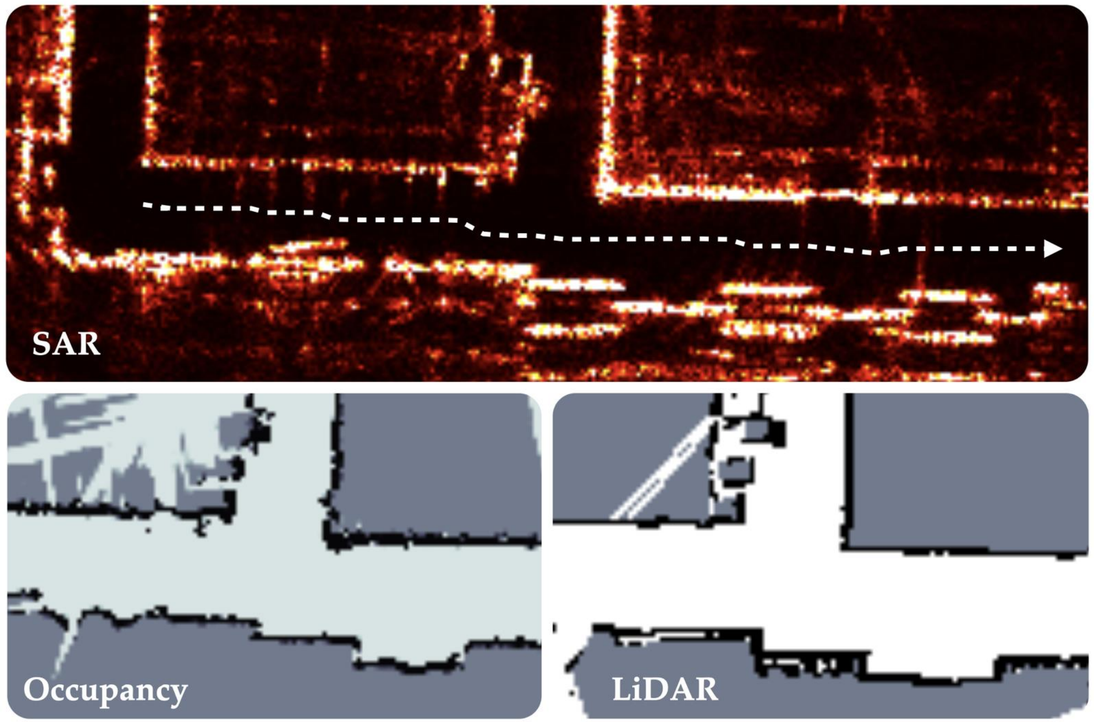




Our Approach

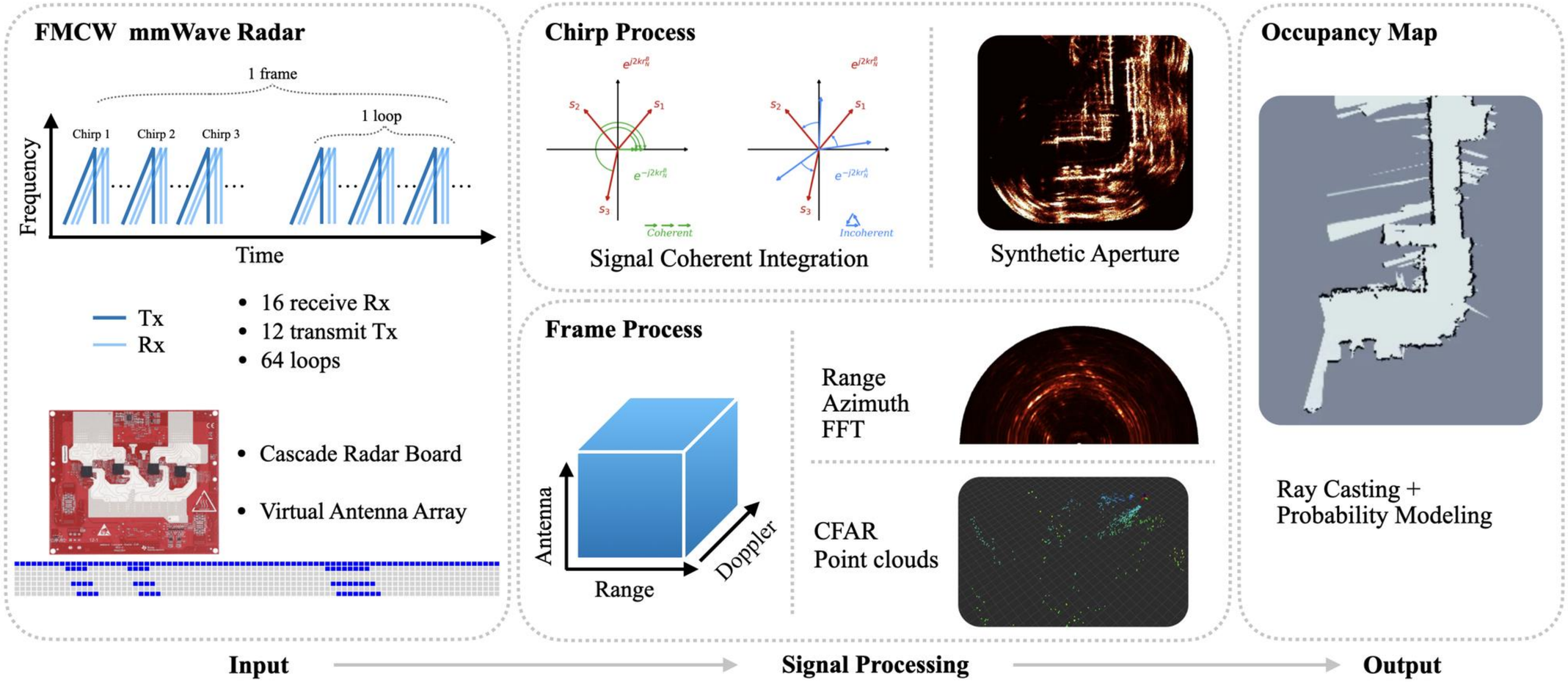
Turn the robot's motion into a giant antenna

- Synthetic aperture: coherently integrate every chirp along the robot path
- Needs accurate antenna poses, which come from odometry
- A probabilistic model turns the SAR image into an occupancy map
- Open dataset (~1.1 TB) + GPU code: github.com/rpl-cmu/rpm



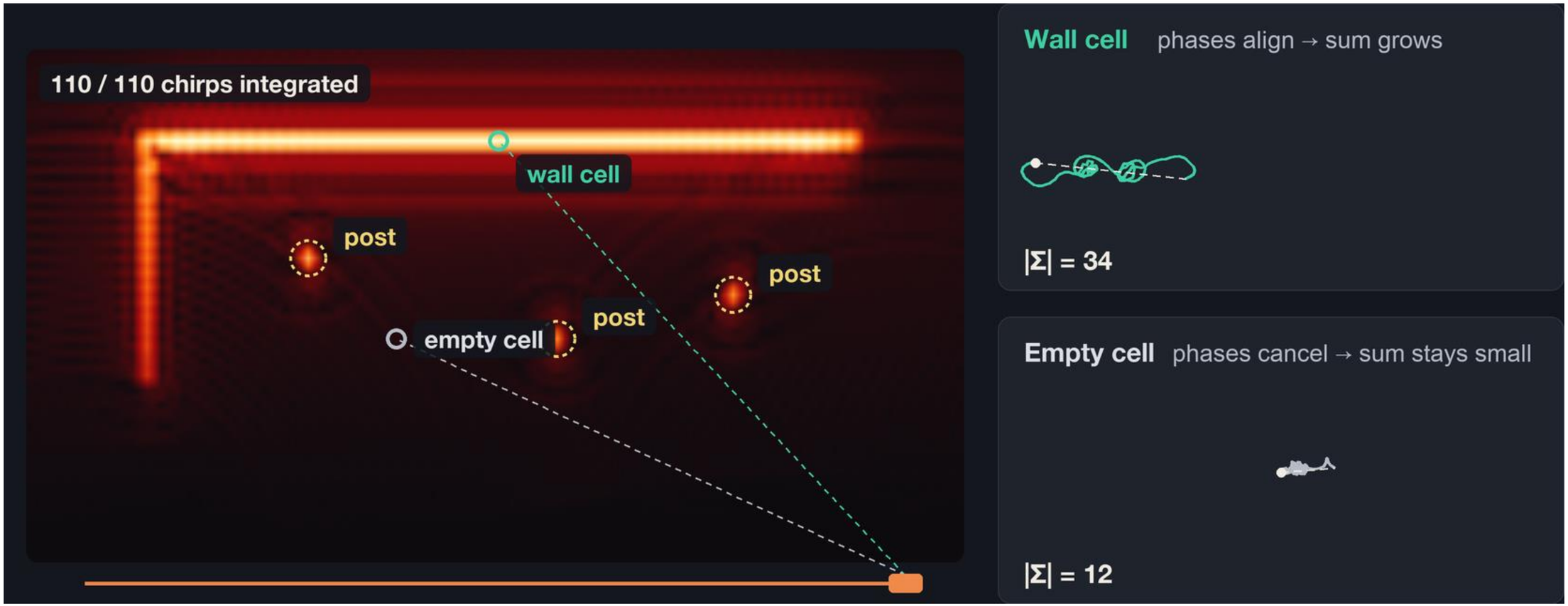
Pipeline

From raw chirps to an occupancy map



Watch the Aperture Focus

Back-projection from 09/03, now on a moving robot



Phases at a wall cell align and the sum grows; at an empty cell they cancel.

Probabilistic Mapping

From SAR amplitude to occupancy probability

1 · Rayleigh CDF

$$R = 1 - \exp(-(|M|/M_n)^2 \,/\, 2{\sigma_a}^2)$$

Normalize the SAR amplitude, then map it to an occupancy probability. $\sigma_a \approx$ radar noise floor.

2 · Ray-cast transmittance

$$T(n) = \prod (1 - R) \text{ along the ray}$$

Cells behind strong returns stay unknown, not free. Also suppresses multipath ghosts.

3 · Log-odds fusion

$$L(n) \mathrel{+}= T(n) \cdot \log(R \,/\, (1 - R))$$

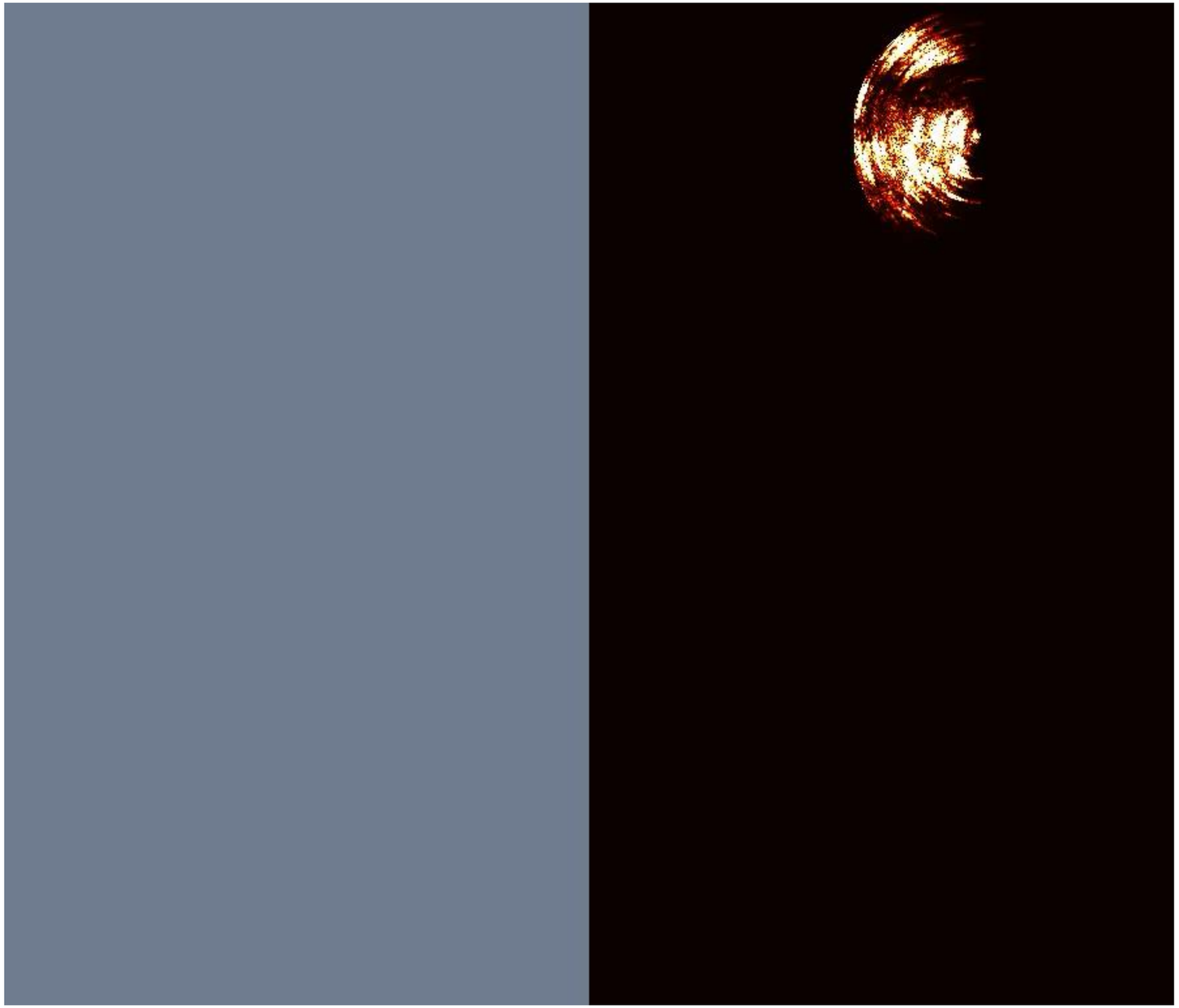
OctoMap-style clamped update keeps the map stable yet quick to adapt.

Only focused SAR updates the map: a two-frame delay and a 30° update cone keep blurry regions out.

SAR Mapping

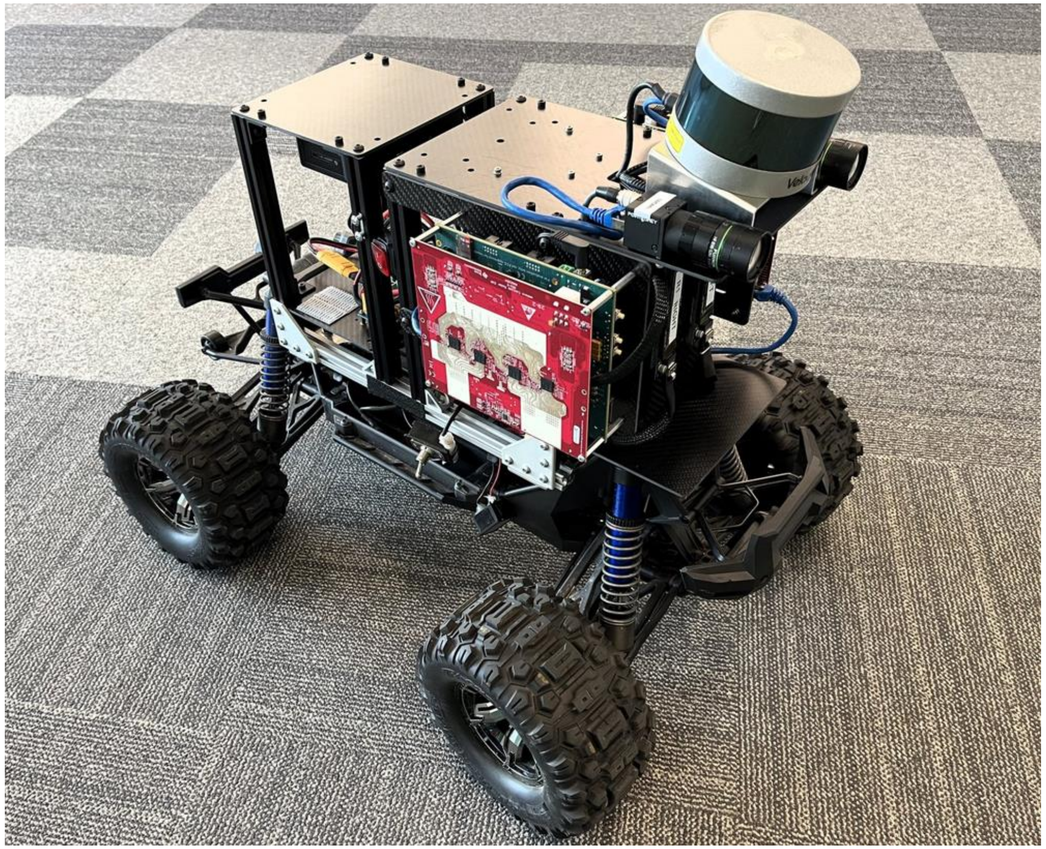
CMU CIC

- Left: occupancy map
- Right: SAR image
- Only focused SAR updates the map



Experiments

Two cascade radars on a ground robot

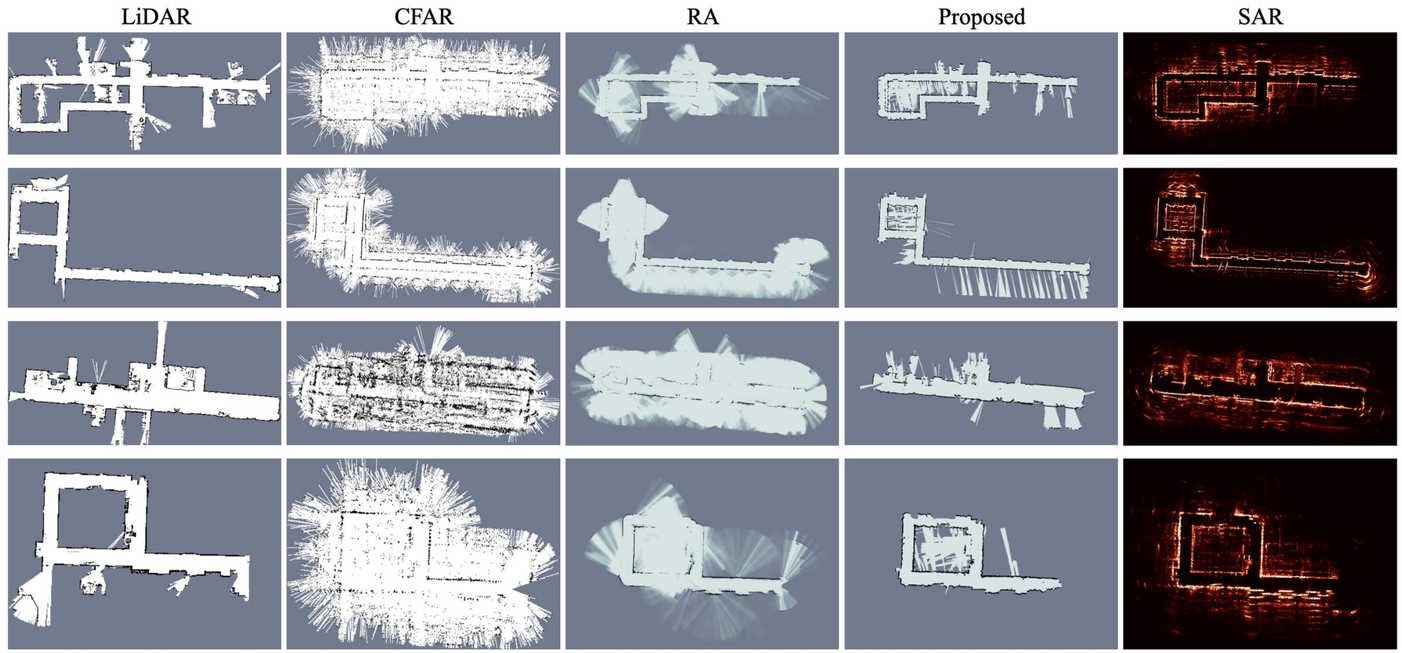


RADAR	2 × TI MMWCAS cascade, one per side, 10 Hz
ANTENNAS	12 Tx × 16 Rx = 192 pairs per radar
SIGNAL	77 GHz · 15 m range · 6 cm resolution
REFERENCE	VLP-16 LiDAR, stereo cameras, IMU
POSES	FAST-LIO / OpenVINS, interpolated per chirp
DATA	10 indoor sequences: corridors, open workspaces

Map Quality

Only the synthetic aperture keeps the walls

- CFAR: per-frame detections → point cloud → OctoMap
- Range-Azimuth (RA): per-frame image → same mapping steps
- Proposed: chirp-level back-projection along the path → same mapping steps
- Ground truth: LiDAR OctoMap, 0.1 m cells, same poses



Results

Better maps, better paths

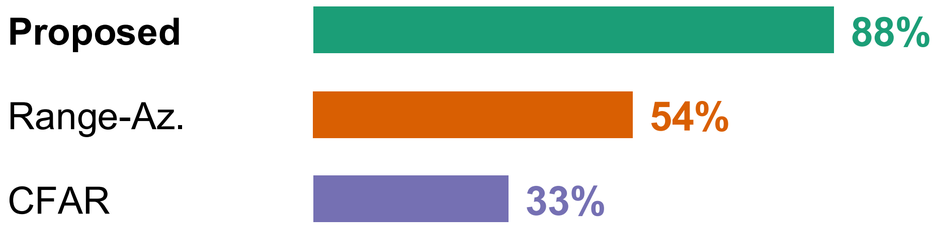
Method	Chamfer ↓	F-score ↑
CFAR	2.31	0.33
Range-Azimuth	1.17	0.60
Proposed	**0.86**	**0.65**

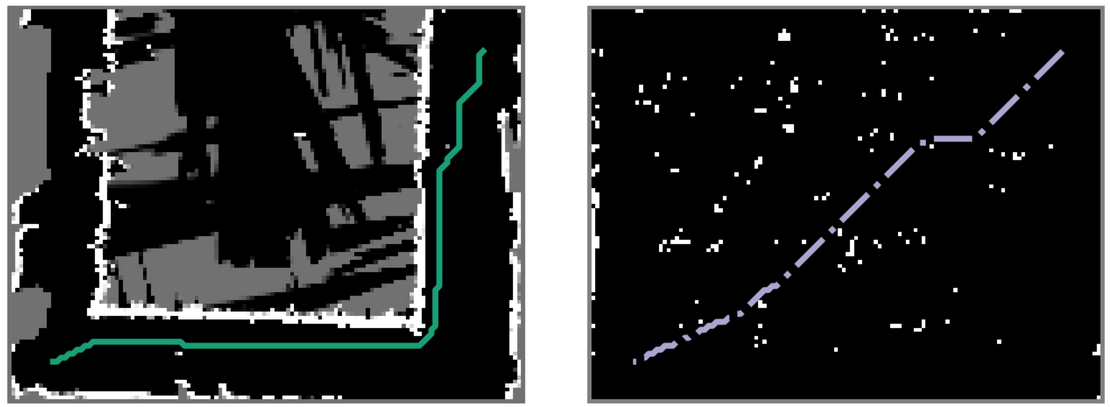
−27%

Chamfer distance vs. the best baseline

Mean over 10 sequences · occupied = p > 0.6 · F-score at 0.2 m

Radar-planned paths that are collision-free on the LiDAR map

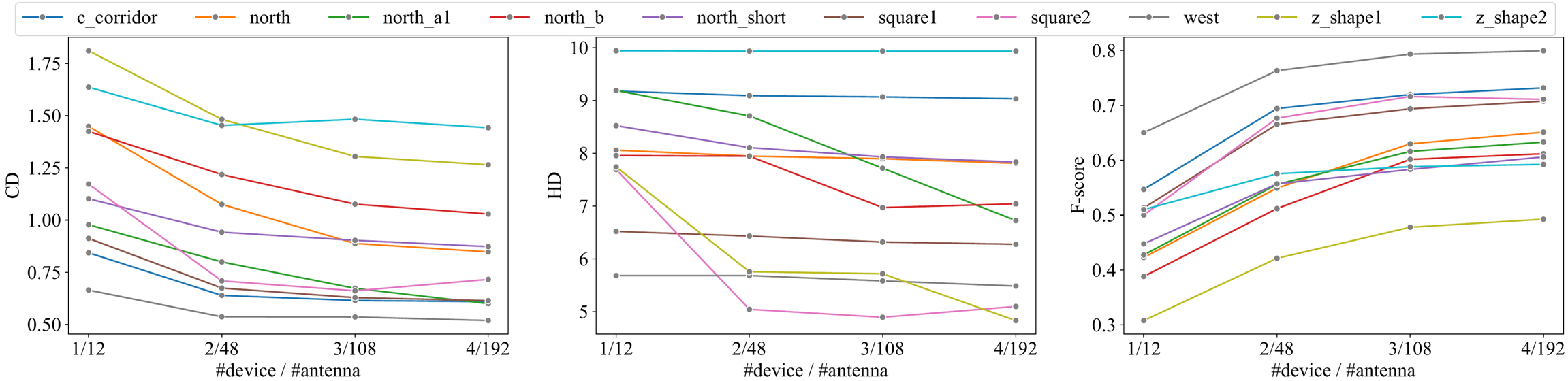




Proposed **CFAR**

Ablation

Three of four radar chips are enough



Chamfer distance (CD), Hausdorff distance (HD) and F-score vs. number of radar chips / antenna pairs, per sequence

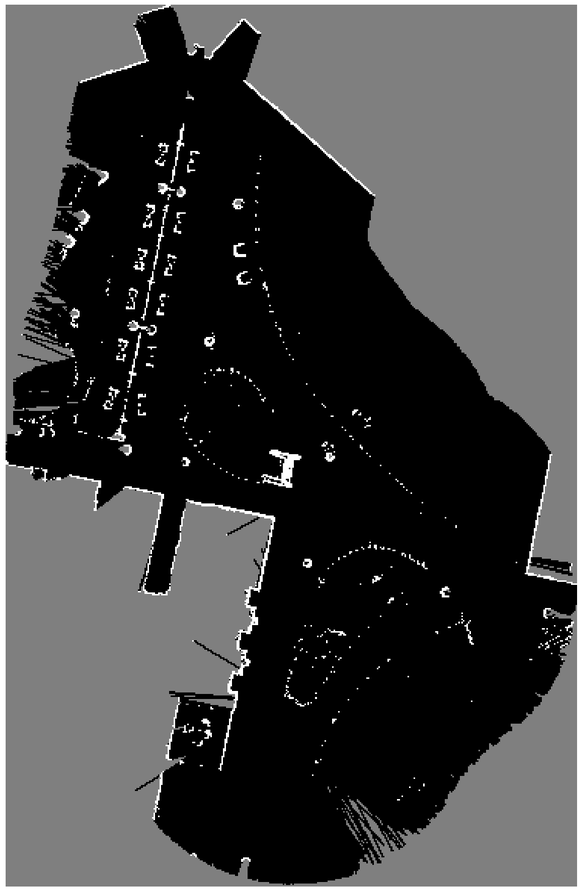
- 12 → 108 antenna pairs: map quality improves steadily with each added chip
- 108 → 192: minor gains, since one board already spans an aperture, so compute can be cut

Limitations

Where radar helps, and where it still struggles



Proposed (radar)



LiDAR

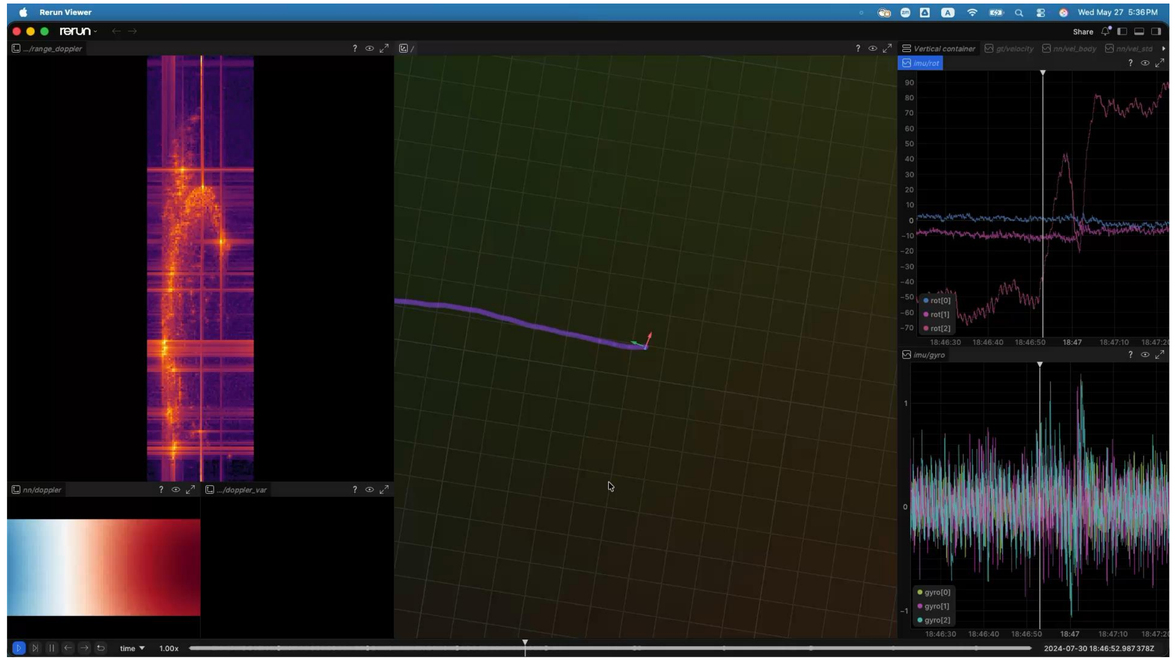
Helps

- Maps glass railings that LiDAR barely sees
- No "mirror world" from glass reflections

Struggles

- Pure rotation: too little aperture
- Needs accurate, well-timed antenna poses, so good odometry matters

- # UNRIO: Uncertainty-Aware Velocity Learning for Radar-Inertial Odometry

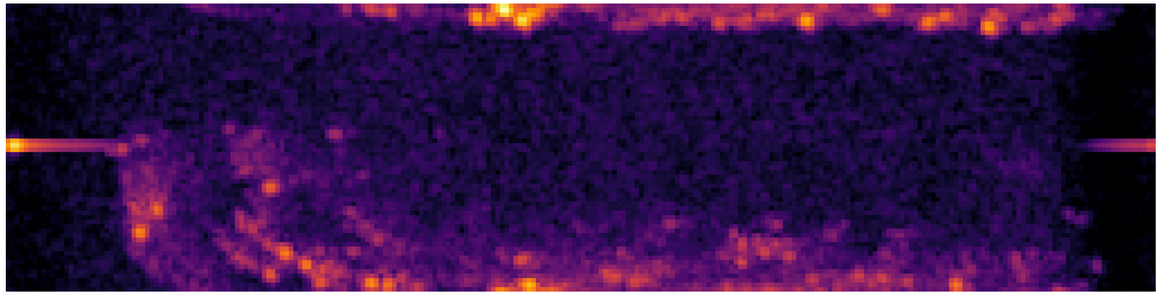


What a Point Cloud Throws Away

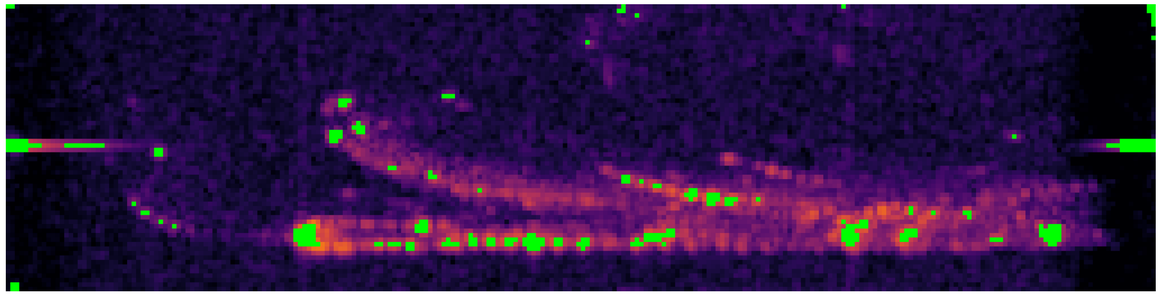
CFAR keeps only a few percent of the spectrum

- Point-cloud RIO only sees CFAR peaks: 1.8% of the bins in this frame
- Ego-velocity is a property of the whole spectrum, observable even without one strong return
- Aliasing: at 1.45 m/s the return folds past the ±1.22 m/s limit, so peak Doppler is biased
- Can we estimate velocity from the full spectrum instead?

Aliased: 1.45 m/s outruns the ±1.22 m/s interval



CFAR keeps only the green bins

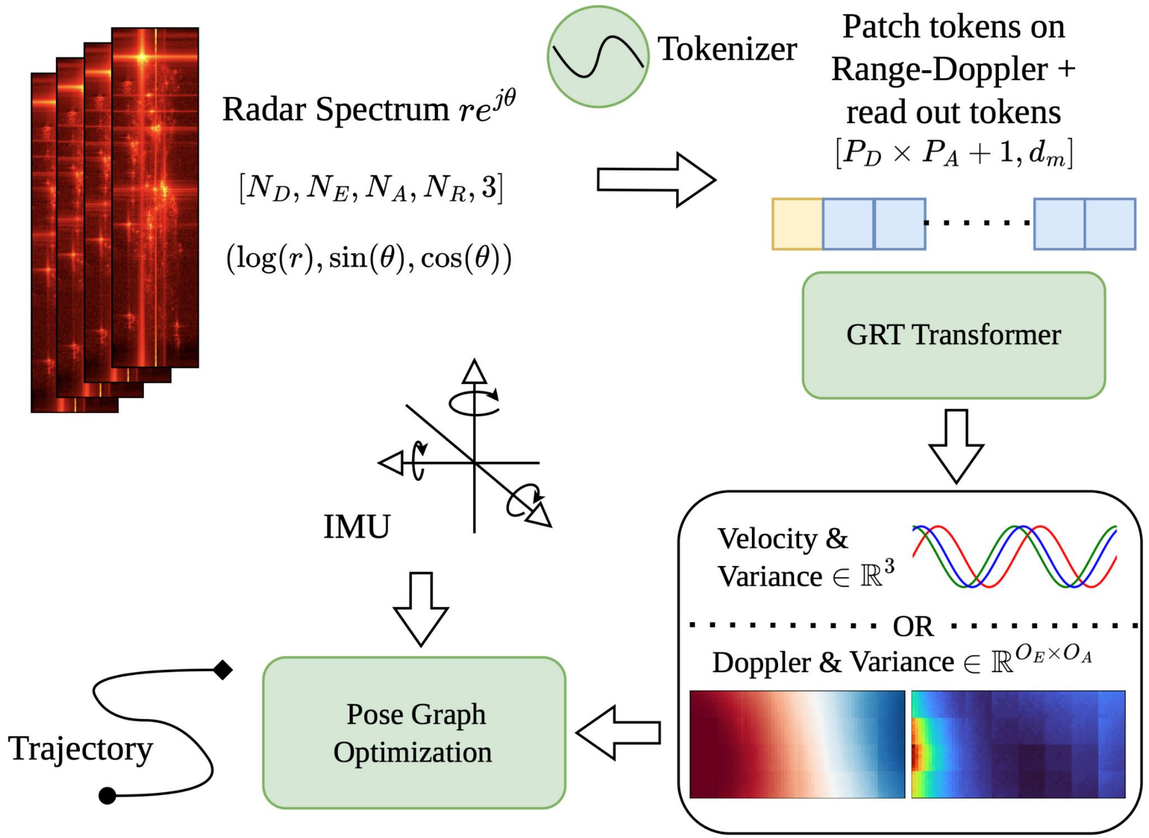


Raw IQ1M range–Doppler spectrum: range 0–11.2 m (horizontal), Doppler ±1.22 m/s (vertical)

Learning Velocity from the Spectrum

Built on GRT, the radar foundation model

- Input: full complex range–Doppler spectrum, encoded as (log r, sin θ, cos θ)
- Tokenize range–Doppler patches + a readout token
- Transformer backbone with multistage training (GRT) [2]
- Two read-outs: velocity + uncertainty directly, or a Doppler image + uncertainty



[2] Huang, Tianshu, et al. "Towards foundational models for single-chip radar." CVPR 2025

Uncertainty-Aware Radar-Inertial Odometry

Learned velocity + covariance become a factor in the pose graph

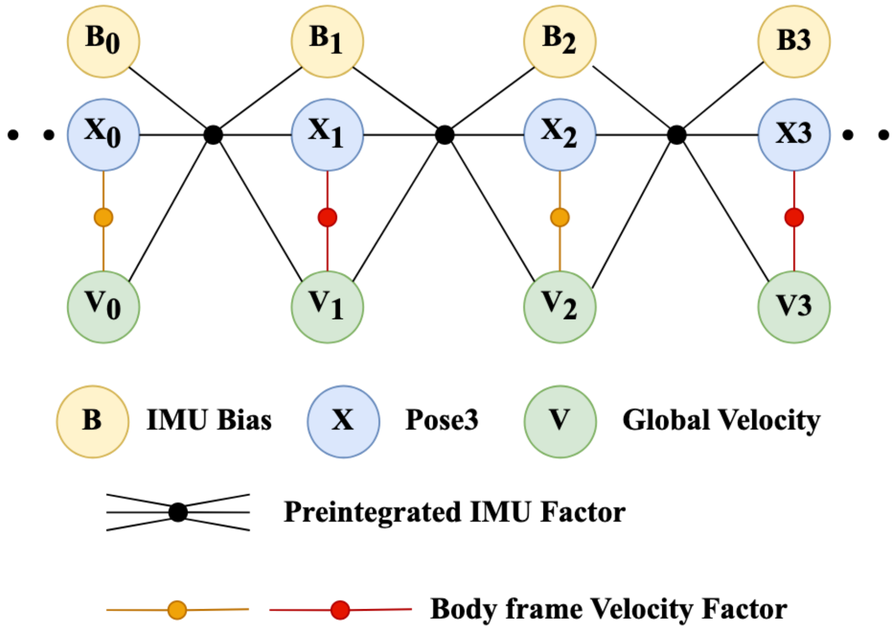
- Train the uncertainty with a negative log-likelihood loss [3]

$$\mathcal{L}_{\text{NLL}} = \lambda_{\text{nll}} \sum_i \frac{(est_i - gt_i)}{2\hat{\sigma}_i^2} + \frac{1}{2} \log \hat{\sigma}_i^2$$

- Doppler read-out: velocity by weighted least squares; covariance from the normal equations

$$\hat{\mathbf{v}}_d = \arg\min_{\mathbf{v}} \sum_{e,a} w(e,a) \left[\hat{\mathbf{d}}_m(e,a) - \mathbf{v} \cdot \hat{\mathbf{u}}(e,a)\right]^2$$

- Sliding-window pose graph: radar-velocity factor weighted by Σ + IMU preintegration [1]
  - Same least-squares machinery as slide 13

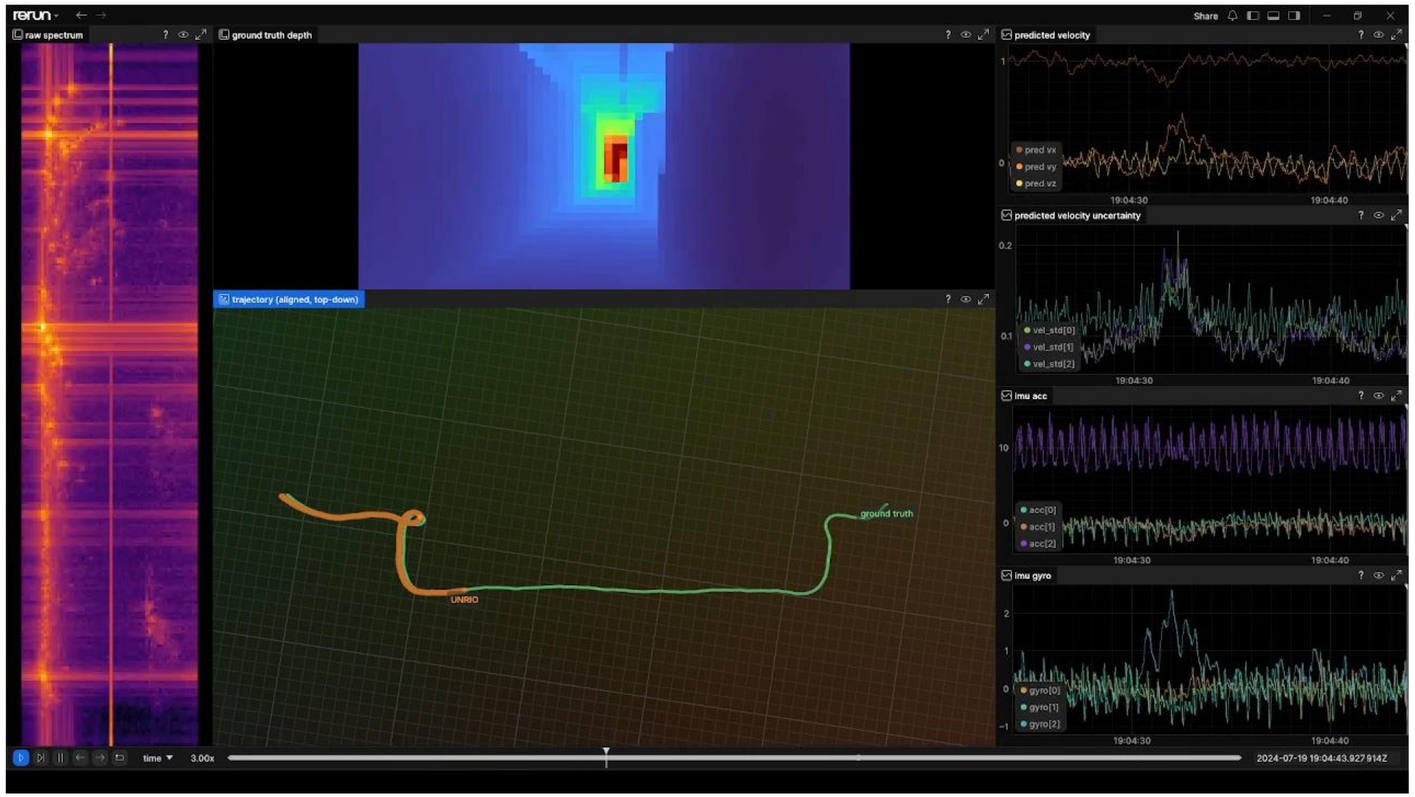


[1] Huang, Jui-Te, et al. "Multi-radar inertial odometry for 3D state estimation using mmWave imaging radar." ICRA 2024 · [3] Qiu, Yuheng, et al. "MAC-VO: Metrics-aware covariance for learning-based stereo visual odometry." ICRA 2025

Results: Forward Motion

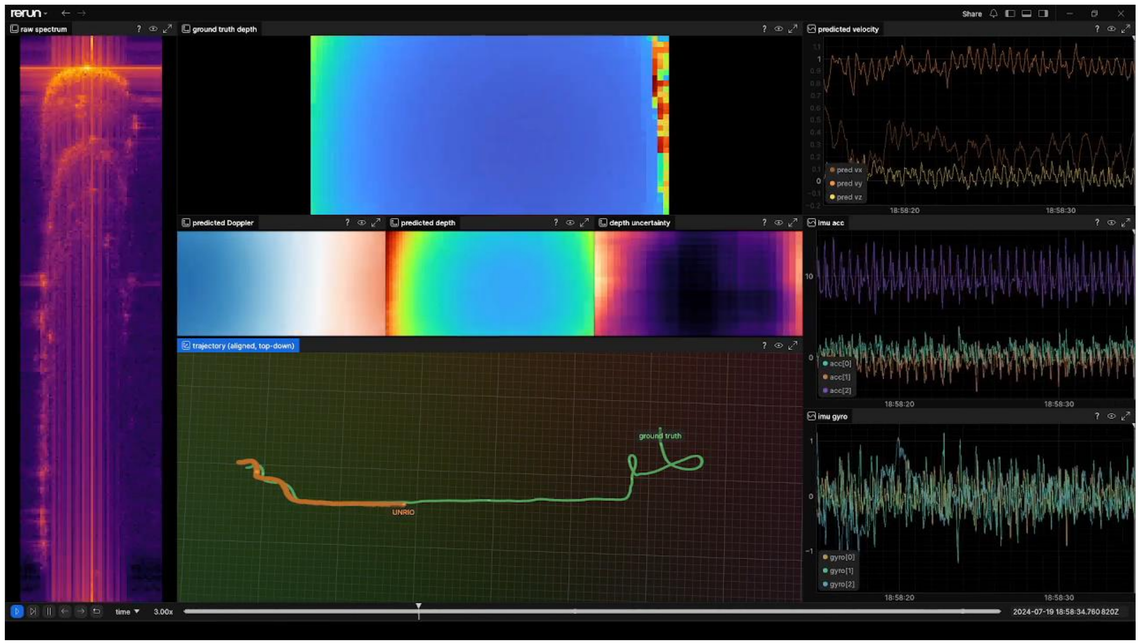
Velocity read-out on a held-out indoor corridor sequence

- Trained on 29 hours of raw complex radar data [2]
- Labels: LiDAR-inertial odometry pseudo ground-truth body velocity
- Tested on unseen sequences
  - Clip: input spectrum, network outputs vs. LiDAR depth, estimated trajectory

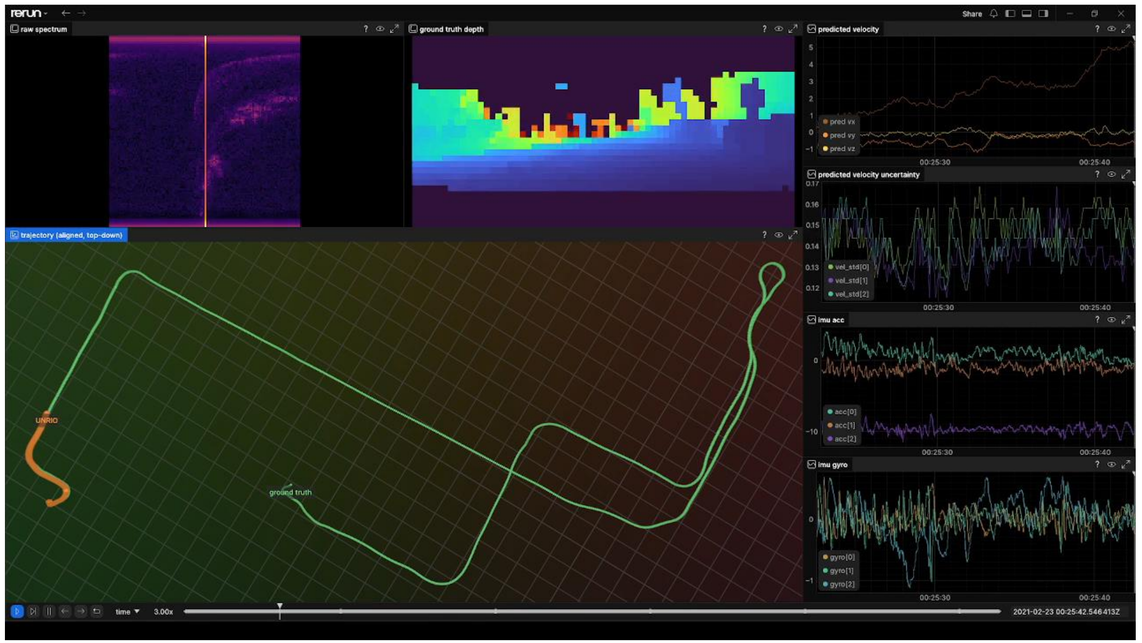


Results: Lateral Motion and High Speed

Where point-cloud methods struggle



IQ1M, Doppler read-out under lateral motion: note the sign change across azimuth

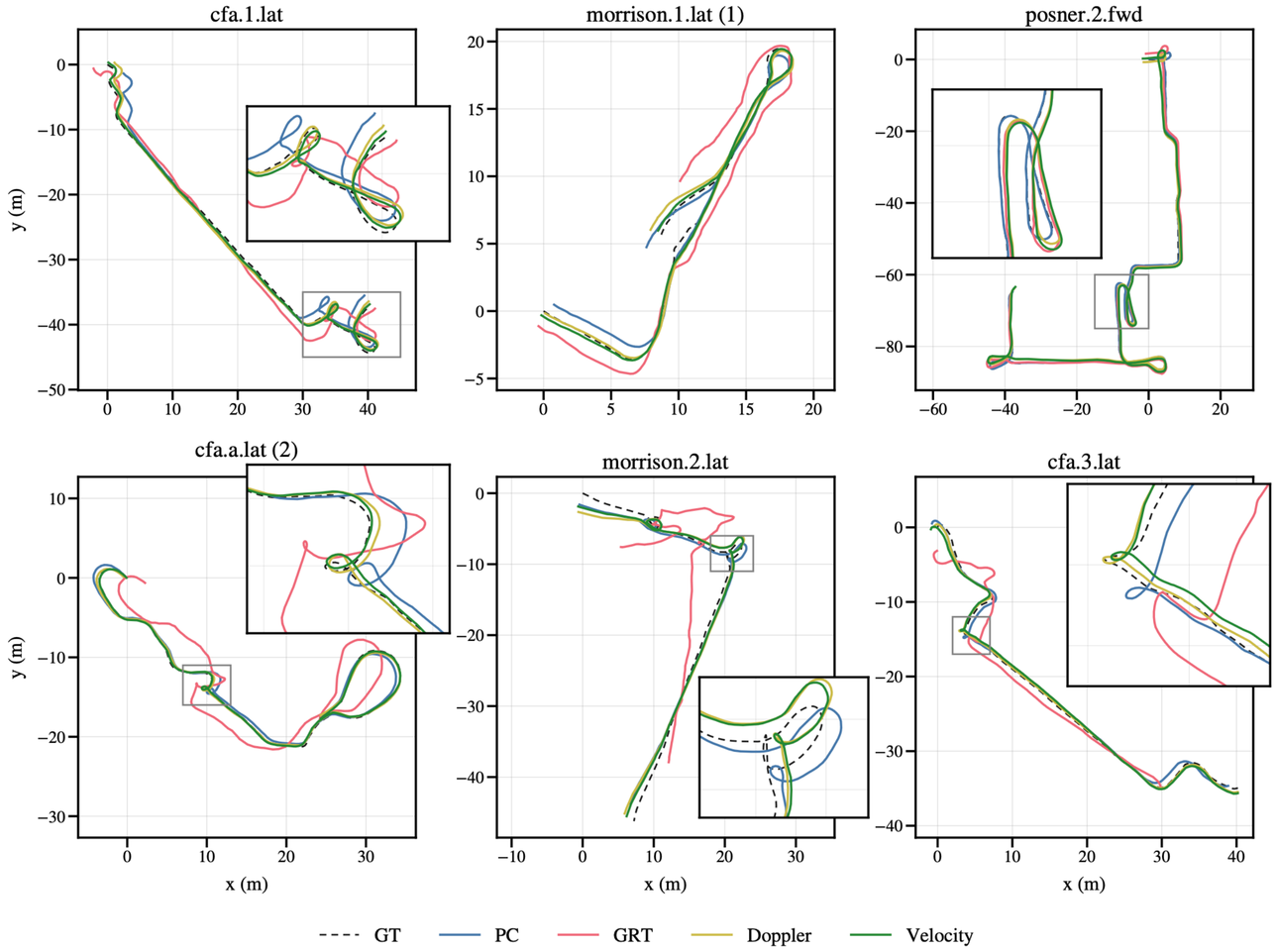


ColoRadar longboard at 4–8 m/s, past the ±2.50 m/s Doppler limit: the learned velocity stays bounded

Evaluation

Lowest relative pose error on most sequences

- Best performance on most forward-motion traces
- Best performance on lateral-motion traces
- Holds across datasets, motion patterns, and radar modulations
- Runs in real time



References

My work and other works cited in this talk

My work

- J.-T. Huang, R. Xu, A. Hinduja, M. Kaess. "Multi-Radar Inertial Odometry for 3D State Estimation using mmWave Imaging Radar." ICRA 2024.
- J.-T. Huang*, R. Xu*, M. Kaess. 4D Radar SLAM Challenge entry, ICRA 2026. (*equal contribution)
- J.-T. Huang, R. Xu, M. Kaess. "Millimeter Wave Radar: From Synthetic Aperture to Probabilistic Mapping." IROS 2026. github.com/rpl-cmu/rpm
- J.-T. Huang, T. Huang, A. Rowe, M. Kaess. "UNRIO: Uncertainty-Aware Velocity Learning for Radar-Inertial Odometry." Under review.

Other references

- S. Zhao, P. Wang, H. Zhang, Z. Fang, S. Scherer. "TP-TIO: A Robust Thermal-Inertial Odometry with Deep ThermalPoint." IROS 2020.
- Z. Hong, Y. Petillot, S. Wang. "RadarSLAM: Radar based Large-Scale SLAM in All Weathers." IROS 2020.
- C. Doer, G. F. Trommer. "x-RIO: Radar Inertial Odometry with Multiple Radar Sensors and Yaw Aiding." Gyroscopy and Navigation, 2021.
- T. Huang, et al. "Towards Foundational Models for Single-Chip Radar." CVPR 2025.
- Y. Qiu, et al. "MAC-VO: Metrics-Aware Covariance for Learning-based Stereo Visual Odometry." ICRA 2025.
- M. Kaess. 16-833 Robot Localization and Mapping, lecture slides, Carnegie Mellon University (SLAM and factor graph intro).