

VIREO: Visual-Inertial-based Reconstruction with Euclidean-Distance Optimization

Yuze Du¹, Chuchu Chen², Zijun Sha³, and Guoquan Huang¹

¹ University of Delaware, Newark, DE 19716, USA
{yuzedu, ghuang}@udel.edu

² The George Washington University, Washington, DC, USA
chuchu.chen@gwu.edu

³ Frontier Research Center, Toyota Motor Corporation, Tokyo, Japan
zijun_sha@mail.toyota.co.jp

Abstract. While recent feed-forward 3D reconstruction models offer a powerful vision-based solution by regressing dense pointmaps from images, their inherent scale ambiguity, cross-frame inconsistency, and unconstrained computational cost hinder long-horizon and/or real-time deployment. This paper presents an online visual-inertial-based reconstruction framework with Euclidean-distance pose optimization that integrates three modules: (i) Visual-Inertial Motion Tracking as frontend (ii) Per-keyframe dense pointmap prediction, with customized keyframe selection, a feed-forward 3D reconstruction network to predict dense pointmap and sparse-to-dense depth alignment to recover metric-scale pointmap, and (iii) Euclidean Distance Field (EDF)-based pose optimization to efficiently refine the current keyframe pose, without analytical (re-)evaluation of distance residuals in optimization iterations. The refined keyframe pose and its uncertainty serve as a global pose measurement fed back to the frontend to reduce long-term drift. Extensive evaluations on the public benchmarking datasets demonstrate the proposed VIREO achieves superior localization accuracy and dense reconstruction fidelity over the state-of-the-art (SOTA) methods. Moreover, we successfully deploy it on a humanoid robot, processing a 5900-frame sequence entirely on a Jetson Orin NX (16 GB) and achieving real-time navigation with approximately 10 FPS dense reconstruction on selected keyframes.

Keywords: Visual-Inertial SLAM · 3D Computer Vision · Robot Vision

1 Introduction and Related Work

Online 3D reconstruction provides essential spatial awareness for robotic tasks like motion planning and environment perception [12]. While active depth sensors are common, they face fundamental limitations: RGB-D cameras degrade outdoors, and LiDARs are expensive, bulky, and lack high-resolution textures. Consequently, high-fidelity online dense 3D reconstruction using only vision (and proprioception) has become a critical pursuit in embodied AI.

Feed-forward 3D networks have recently emerged to directly infer visual geometry, bypassing the feature-matching steps of traditional pipelines. As pioneering efforts, DUS3R [27] and MAS3R [13] demonstrate zero-shot inference of dense pointmaps and correspondences from image pairs. Later, VGGT [25] generalized this to multi-view joint inference of scene geometry and poses. To enable sequential online processing, recent models such as StreamVGGT [30], CUT3R [26], and TTT3R [7] have been introduced. Pre-trained on diverse datasets, these models excel at delivering high-fidelity local structural consistency. Despite their impressive geometry generation capabilities, these pure neural networks expose bottlenecks when deployed over long-term trajectories and edge devices, in particular, suffering from severe scale ambiguity and out-of-memory issues. Without robust and efficient pose estimation, sequentially generated per-frame pointmaps fail to align accurately, resulting in ghosting artifacts that cause the accumulated map virtually unusable.

Recent advances in visual-inertial navigation systems (VINS) [11]—which fuse camera and Inertial Measurement Unit (IMU), providing metric-scale acceleration and angular velocity measurements, offering accurate, efficient, and robust metric motion tracking—should be leveraged to support 3D reconstruction on scale ambiguity and out-of-memory issues. From the perspective of iterative updates and relinearization, existing VINS frameworks can be broadly categorized into optimization-based methods [3, 5] and filtering-based methods [1, 6, 9, 16, 17]. The former rely on iterative relinearization and sliding-window bundle adjustment for high accuracy, while the latter perform one-time linearization for real-time efficiency. Despite accurate pose estimation, standard VINS only yield sparse landmarks, falling short of dense 3D reconstruction.

Furthermore, to enhance geometric consistency of adjacent keyframes and further reduce the motion drift induced by VINS, we require fast, efficient pose optimizer through point cloud registration. Although conventional point cloud registration algorithms (e.g., [21], [2] and [15]) may achieve this, the speed is slower than 1FPS while we need 30-40 FPS for online 3D reconstruction.

Therefore, in this paper, we propose VIREO, to utilize the commonplace monocular camera and six-axis IMU as the only data source and integrate a VINS-based motion tracker and feed-forward 3D networks for metric-scale online 3D construction. Moreover, we develop an efficient pose optimizer by aligning the keyframe pointmaps via a novel Euclidean Distance Field (EDF)-based pose optimization, *without* explicitly searching feature correspondences and analytically (re)evaluating residuals. The optimized keyframe pose estimate along with its uncertainty is quickly fed into the VINS frontend as a global measurement to suppress potential localization drifts. The resulting pose estimates in turn improve 3D online reconstruction over time. As a result, the proposed VIREO is able to achieve real-time navigation alongside approximately 10 FPS dense reconstruction on selected keyframes, when validated on the platforms with limited computing power, such as Jetson Orin NX (16GB) and RTX 2000 Ada GPUs.

In particular, the main contributions of this paper include:

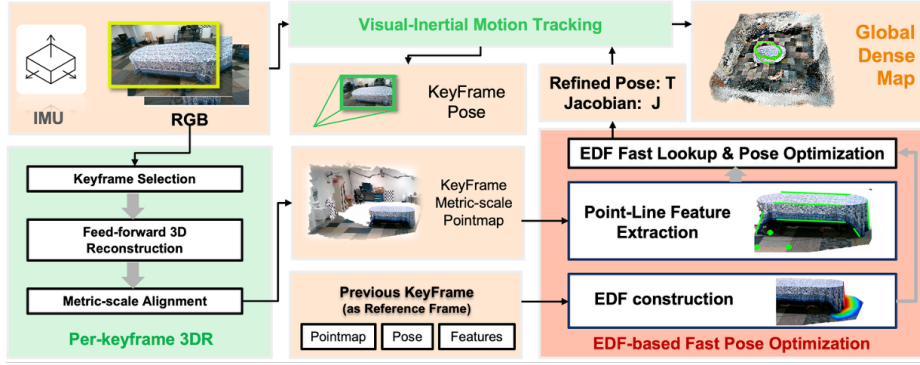


Fig. 1: The proposed VIREO system architecture.

- We propose an online monocular visual-inertial-based reconstruction system with Euclidean-distance optimization (VIREO) by efficiently using VINS as frontend motion tracking and a novel EDF-optimized 3D reconstruction backend, to produce metric-scale, geometrically consistent 3D point cloud and accurate motion tracking.
- We develop a novel EDF-based pose optimization to efficiently align the scaled 3D keyframe pointmaps, without explicit search of geometric feature correspondences and analytical (re)evaluation of residuals.
- We perform extensive evaluations showing that the proposed VIREO significantly improves pose estimation accuracy and metric fidelity while strictly maintaining real-time reconstruction performance onboard (10 FPS on selected keyframes).

2 VIREO System Overview

In this section, we provide an overview of our system design. As illustrated in Fig. 1, the proposed VIREO architecture operates through three core modules:

1. *Visual-Inertial Motion Tracking*: A robust and efficient $\sqrt{\text{VINS}}$ [17] frontend provides real-time pose estimation along with metric-scale sparse 3D landmarks that are used to align the feed-forward 3D dense pointmaps.
2. *Per-Keyframe Dense Metric Reconstruction*: A feed-forward network predicts dense pointmap for each keyframe in a single forward pass, and pointmap is immediately anchored to a consistent metric scale using landmarks from visual-inertial tracking module.
3. *EDF-based Optimization*: We extract point/line features from the metric-scale pointmap of current keyframe. Subsequently, we build a discrete EDF around each line feature in (previous) keyframe by explicitly encoding the point-to-line distances (residuals) of all the involving voxel grids. By projecting the features from current frame into EDF, querying this EDF directly

quantifies the inter-frame geometric feature misalignment. As such, the current keyframe pose can be efficiently refined by minimizing this misalignment (EDF residuals) without analytical (re-)evaluation of residuals. The optimized pose $\mathbf{T}$, alongside its Jacobian from optimizer $\mathbf{J}$, is fed back to the VINS frontend to bound the navigation drift.

Through this frontend–backend interaction, pose estimation improves dense reconstruction with metric scale, while dense geometric constraints in turn suppress motion tracking drift, jointly enhancing the 3D reconstruction quality. Next, we will explain in detail the main three modules.

3 Monocular Visual-Inertial Pose Estimation

To obtain efficient and consistent pose estimation under resource-constrained deployment, we adopt the $\sqrt{\text{VINS}}$ [17]. It fuses visual-inertial measurements to provide sparse VINS landmarks, metric camera poses and the pose covariance. Specially, the Square-Root Filter (SRF) formulation of $\sqrt{\text{VINS}}$ [17] maintains an upper-triangular square-root covariance matrix $\mathbf{U}$ to improve robustness and efficiency. The uncertainty of the state estimate is represented by a marginal covariance, computed by $\mathbf{P} = \mathbf{U}^\top \mathbf{U}$. We quantify the confidence of each keyframe pose using the square root of the trace of its marginal pose covariance and pass the resulting scalar to the backend to guide geometry refinement.

Also, once a refined keyframe pose is available from the backend EDF optimizer, we treat it as a global pose measurement and integrate into VINS pose update via SRF. Detailed derivations of SRF update are in Appendix B.

4 Per-keyframe Dense Metric Reconstruction

We now explain how the keyframe pointmap is learned and aligned to sparse VINS features to build metric 3D map.

Photometric and Metric-Aware Keyframe Selection: Designed for better reconstruction quality, we combine photometric discontinuities and image structure consistency between adjacent keyframe into photometric consistency score S .

$$S = \frac{\alpha e_{photo} + (1 - \alpha) e_{ssim}}{N_{overlap}} \quad (1)$$

where $N_{overlap}$ denotes the number of overlapping pixels between the two frames, $e_{photo}, e_{ssim} \in [0, 1]$ denote the photometric difference and structural similarity error respectively over the overlapping region. The weight α balances intensity consistency and structural similarity. In practice, we set $\alpha = 0.8$ to penalize illumination changes. We evaluate the score for each incoming frame image against the previous keyframe. A frame is promoted to a keyframe when the score goes above some threshold, then the RGB image is passed to feed-forward 3D network.

Feed-Forward 3D Network For each selected keyframe, we adopt frozen weights from CUT3R [26] as the feed-forward pointmap predictor in the backend. Its local refinement design aligns well with our sequential processing paradigm, offering incredible performance for online inference. While the original CUT3R design maintains learnable pose tokens interacting with a global memory buffer, in our design, the motion tracking frontend estimates metric-scale pose estimates for global trajectory consistency. We thus remove CUT3R’s global memory module and reinitialize the pose token for each keyframe, retaining only the local refinement pathway. As a result, this critical re-design significantly reduces both the cross-attention overhead and memory footprint. The network predicts a dense unscaled pointmap in local camera frame.

Metric-Scale Alignment To ensure the pointmap geometry complies with the physical sensor model, we reconstruct each 3D point by unprojecting its predicted depth z_i^{net} via the pre-calibrated camera intrinsics $\mathbf{K}$:

$${}^C\mathbf{p}_i^{net} = z_i^{net}\mathbf{K}^{-1}[u_i, v_i, 1]^\top \quad (2)$$

We estimate a *Sim*(3) transformation $(s, \mathbf{R}, \mathbf{t})$ by minimizing the misalignment.

5 EDF-based Fast Pose Optimization

Using the geometric features of the current and previous keyframes, we aim to formulate pose constraints that efficiently refine the current keyframe pose, subsequently enhancing the global consistency of reconstructed pointcloud. Traditional point cloud registration algorithms (e.g. [21], [2] and [15]) typically take a few seconds to process dense point cloud registration, while we only have millisecond time window during online reconstruction. Instead, we leverage representative 3D structural lines and minimal sets of anchor points as features for fast point cloud registration, and construct a discretized EDF for fast misalignment evaluation. The EDF is represented as a spatial voxel grid that encodes the analytical distances from voxel centroids to 3D line features within the reference frame. By evaluating line misalignment through *constant-time EDF look-ups* during Levenberg-Marquardt (LM) optimization, our approach eliminates the need for expensive analytical distance (residual) computations, enabling rapid pose refinement.

5.1 Point/Line Features Extractor

Using both depth and RGB modalities, we extract structural line features, alongside a (minimal) set of complementary point features that provide extra constraints. Specifically, from the depth map separated from pointmap, line segments are extracted with the Line Segment Detector (LSD) [10], which are aligned with geometric edges and depth discontinuities. Each line segment is augmented with an RGB-based appearance descriptor for cross-frame matching. As line features cannot constrain motion along the line direction (which has no

effect on the point-to-line distance), we additionally extract a small number of ORB keypoints [19] outside the detected line regions from the RGB images.

5.2 Voxelized EDF Construction

We recall the pointmap, camera poses, and point/line features of the previous frame as the reference frame. We build a bounded 3D volume around each line feature from the reference frame (i.e. reference lines) and discretize it into cubic voxels. Each voxel stores the analytical Euclidean distance from each voxel centroid to the reference line, constructing a discrete scalar field (shown in Fig. 2). However, fixed-size grids struggle to balance efficiency and spatial coverage: high VINS pose uncertainty can push current-frame features outside small grids (causing look-up failures), whereas overly large grids induce unnecessary computational overhead. To address this inherent efficiency-sufficiency trade-off, we introduce an adaptive spatial expansion mechanism by querying the pose uncertainty from motion tracker (Section 3). Since higher pose uncertainty naturally implies larger spatial deviations between matched lines, we utilize the square root of the pose covariance trace to adaptively expand the bounding box.

5.3 EDF Residual and Pose Optimization

We build pose-dependent geometric feature measurement residuals of line features by leveraging pre-computed EDFs. To construct a continuous and differentiable optimization problem while avoiding the alignment ambiguity caused by relying solely on endpoints, we discretize these continuous line segments into a sparse set of 3D sample points, ${}^C\mathbf{p}_i \in \mathbb{R}^3$, at intervals of half the voxel resolution. Defined in the local camera frame, these points are subsequently transformed into the global frame using the current pose estimate $\mathbf{T} \in SE(3)$: ${}^W\mathbf{p}_i = \mathbf{T} {}^C\mathbf{p}_i$. We then evaluate the distance for each sample point ${}^W\mathbf{p}_i$ by identifying the specific voxel cell in which it resides. For a given pose estimate $\mathbf{T}$ and a sample point ${}^W\mathbf{p}_i$, its residual is thus instantly queried as the *pre-computed* distance stored at its nearest voxel:

$$r_i^{EDF}(\mathbf{T}) = \mathcal{D}({}^W\mathbf{p}_i) \quad (3)$$

where $\mathcal{D}(\cdot)$ is the EDF value stored in the voxel cell in which ${}^W\mathbf{p}_i$ resides. Summarizing residuals for all sample points on current frame lines yields the line misalignment residual.

To resolve the sliding ambiguity of lines, we incorporate the complementary point features extracted. The measurement residual for each point feature that describes points deviation $r_j^{pts}(\mathbf{T})$ is the 3D Euclidean distance to its matched counterpart in the reference frame. Because these point features are strictly capped in quantity and serve purely as complementary constraint, computing their residuals and Jacobians incurs negligible computational overhead. Finally, the full pose optimization is given by:

$$\mathbf{T}^* = \arg \min_{\mathbf{T}} \left(\underbrace{\sum_i \rho(r_i^{EDF}(\mathbf{T})^2)}_{\text{Line Feature Misalignment Term}} + \underbrace{\sum_j \rho(r_j^{pts}(\mathbf{T})^2)}_{\text{Point Feature Misalignment Term}} \right) \quad (4)$$

and $\rho(\cdot)$ again is the Huber loss.

The Jacobian derivations of the line misalignment are provided in Appendix A. An iterative Levenberg–Marquardt algorithm is used to solve this problem, and the refined pose $\mathbf{T}^*$ and the Jacobian $\mathbf{J}$ of the LM optimizer are fed back to the motion tracker (Section 3) as a global measurement. Analytic SRF information matrix update with the global measurement is shown in Appendix B.

5.4 Remarks

We point out that the construction of a high-resolution voxelized EDF introduces an upfront computational overhead. However, calculating the distance for each voxel is entirely independent of any data dependencies, making this process highly parallelizable. Moreover, in each LM iteration, as the camera pose updates, the system must repeatedly re-evaluate cross-products, vector norms, and Jacobians for every sample point. Crucially, calculating the exact Euclidean distance inevitably involves frequent calls to the square root (`sqrt`) function. At the micro-architectural level, `sqrt` is a highly expensive non-linear operation that consumes significant clock cycles and severely stalls the instruction pipeline.

6 Experimental Results

To validate the proposed VIREO system, three public benchmarking datasets and one custom dataset are used: (i) EuRoC MAV dataset that includes 11 sequences with aggressive UAV motions, (ii) FAST-LIVO2 [29] dataset that has 8 large-scale outdoor sequences enabling validation of our geometric consistency constraints over long trajectories, (iii) RPNG AR Table [4] dataset that contains 8 sequences capturing complex tabletop and indoor structures, providing detailed ground-truth geometry for dense reconstruction evaluation, and (iv) our own “Grand Conference Hall” data collected with Unitree G1 humanoid robot and comprising over 5900 frames to evaluate long-term online deployment.

To ensure a fair and comprehensive comparison of our proposed VIREO, a complete set of SOTA systems including both vision-only and visual-inertial

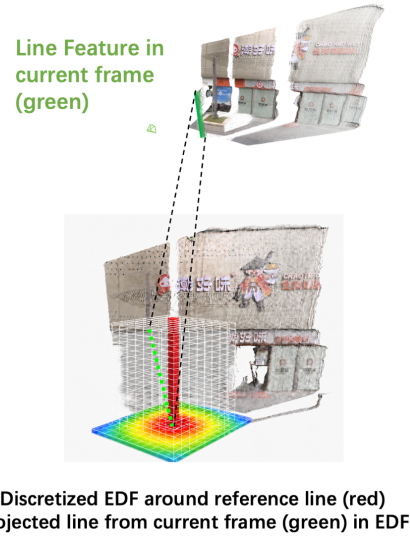


Fig. 2: EDF-based geometric alignment. A discrete EDF voxel grid is constructed around the reference line (bottom) in the global frame. The current line (green) is projected to discretized EDF. Spatial misalignment is efficiently queried via EDF lookup within this grid, directly serving as the geometric residual for pose refinement.

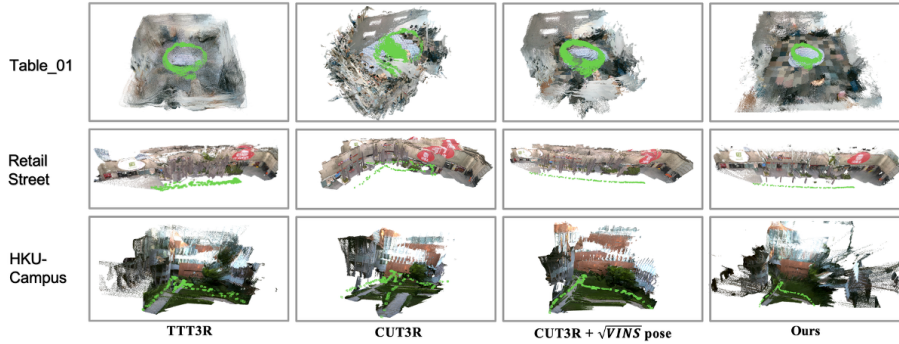


Fig. 3: Qualitative comparison of dense reconstructions. Baseline networks suffer from structural bending and ghosting artifacts over long trajectories, our method produces globally consistent, crisp, and drift-suppressed 3D models.

systems are selected for evaluation. Vision-only systems include classical feature-based SLAM methods: SVO [8] and ORB-SLAM3 [3]; learning-based methods: DROID-SLAM [24]; hybrid dense-mapping systems: Splat-SLAM [20] and HI-SLAM2 [28]; and purely feed-forward 3D reconstruction networks: TTT3R [7] and CUT3R [26]. Note that for unscaled network outputs, a $Sim(3)$ alignment is applied against the ground truth prior to evaluation, whereas VIREO directly operates in metric scale and is evaluated without such scale alignment. Visual-inertial navigation methods include OKVIS [14], VINS-Mono [18], OpenVINS [9], and $\sqrt{VINS}$ [17].

All evaluations of VIREO run on an NVIDIA RTX 2000 Ada Laptop GPU or NVIDIA Jetson Orin NX (16 GB), used as onboard processor of Unitree G1 humanoid robot. To evaluate impact of EDF voxel resolution, we test the impact of different resolutions (VIREO-5/15/25/35cm) on the Absolute Trajectory Error (ATE) for EuRoC MAV dataset, and on runtime efficiency for Grand Conference Hall sequence. In all other experiments, our method (denoted as **Ours**) operates at the default 15 cm resolution (VIREO-15cm). We evaluate pose estimation and dense reconstruction independently. Navigation performance is quantified via Absolute Trajectory Error (ATE) [9]. For reconstruction quality, following iMAP [23], we sample 200,000 points from both meshes to compute three metrics: Accuracy (mean distance from samples in reconstructed mesh to nearest ground-truth points), Completion (mean distance from samples in ground-truth mesh to nearest reconstructed points), and Completion Ratio (points with completion error below 20 cm). Because our datasets span much larger scales than iMAP [23] benchmarks, Replica [22], we adjust the standard 5 cm threshold to 20 cm to reflect performance comparison.

6.1 Motion Tracking Accuracy and Reconstruction Quality

Trajectory Evaluation To evaluate pose estimation (tracking) performance, we compare our VIREO against the state-of-the-art visual or visual-inertial system on EuRoC MAV, RPNG AR Table, and FAST-LIVO2, in terms of ATE. For

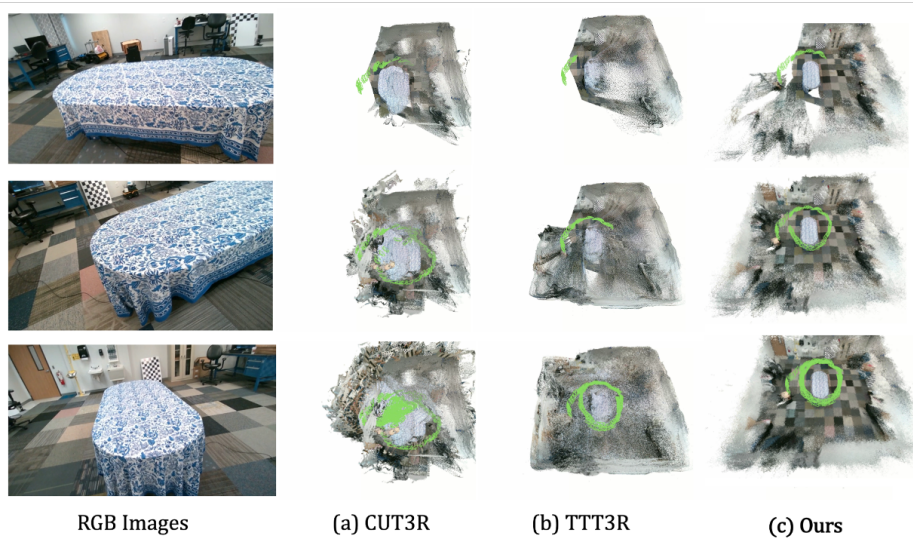


Fig. 4: Test results on RPNG Table-01 Dataset [4], an indoor dataset with repeated loops, at 3 timestamps (begin, first loop, finish). Our 3D reconstruction (c) has better motion tracking and geometric consistency. CUT3R [26] (a) fails midway with ghosting artifacts, while TTT3R [7] (b) produces blurred reconstructions.

brevity and clarity of presentation, the tracking evaluations on the RPNG AR Table and Fast-LIVO2 datasets focus on a curated subset of the most representative and top-performing baselines. As shown in Tables 1-3, purely feed-forward networks (CUT3R, TTT3R) generally exhibit substantially larger trajectory errors than VIREO, particularly on EuRoC and RPNG, while the gap for TTT3R is less pronounced on FAST-LIVO2. Classical methods perform stably but degrade under texture-poor or motion-blurred conditions, while several RGB-only baselines fail or severely drift in FAST-LIVO2’s large-scale outdoor environments. Conversely, VIREO remains robust across all datasets. Notably, Table 1 reveals that decreasing the voxel resolution from 35 cm to 15 cm sharply reduces tracking errors, whereas further refinement to 5 cm yields marginal gains, bounded by the pointmaps’ inherent accuracy. Regardless, VIREO at finer resolutions consistently dominates the top rankings, confirming our framework effectively merges classical VINS robustness with neural structural richness.

Dense Reconstruction Quality We quantitatively evaluate the fidelity of the generated 3D models on the RPNG AR Table dataset using the standard mesh metrics: Accuracy, Completion, and Completion Ratio ($< 20\text{cm}$). We specifically benchmark against HI-SLAM2, CUT3R, and TTT3R, as they represent the SOTA architectures capable of jointly performing visual tracking and dense depth-based reconstruction. As shown in Table 4, the unconstrained feed-forward networks exhibit distinct degradation patterns over long sequences. Due to the lack of global spatial constraints, CUT3R suffers from catastrophic geometric collapse, yielding massive accuracy errors (often exceeding 60 cm) and abysmal Completion Ratios below 3%. While TTT3R mitigates this slightly via global

Table 1: Trajectory Accuracy (Translation(cm)/Rotation(degree)) on the EuRoC MAV Dataset. Top three results in translation and rotation are evaluated independently and highlighted in **Red** (1st), **Blue** (2nd), and **Green** (3rd).

Method	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03											
RGB Input																						
SVO	1.15	10.12	0.94	12.42	2.14	21.24	16.25	29.45	1.24	24.22	1.23	7.80	2.35	8.12	1.21	6.25	1.35	9.45	20.37	11.00	12.25	78.00
Splat-SLAM	15.45	257.6	16.55	266.0	17.65	312.5	24.45	458.1	30.55	360.8	45.35	168.9	48.58	166.6	51.75	128.6	74.15	198.8	68.24	195.8	77.45	190.8
TTT3R	122.4	421.4	49.65	385.9	60.12	293.0	135.4	414.5	70.45	381.0	51.20	155.4	44.35	128.7	55.61	118.0	55.45	141.9	37.15	93.56	37.45	101.8
CUT3R	135.3	643.9	74.12	432.5	74.52	355.0	115.3	651.4	102.1	375.4	19.45	144.7	70.75	238.9	66.52	162.2	42.25	143.3	32.25	74.90	41.35	115.2
ORB-SLAM3	1.31	7.52	0.70	7.68	5.42	36.50	0.98	15.46	10.39	72.12	0.62	4.12	2.42	7.41	2.39	10.46	0.89	6.27	1.01	4.51	0.99	9.31
DRUID-SLAM	2.45	16.30	1.39	12.10	1.82	24.20	1.11	39.90	1.31	27.00	2.56	10.30	2.81	16.50	3.25	15.80	2.45	10.20	3.41	11.50	4.15	20.40
HI-SLAM2	1.35	5.66	0.39	4.44	0.53	3.71	2.56	10.86	3.45	9.07	1.15	4.55	0.75	2.32	1.13	3.49	0.78	5.56	0.89	7.77	1.43	3.92
RGB + IMU Input																						
$\sqrt{VINS}$	1.97	10.25	0.74	14.25	0.88	11.35	0.99	25.18	1.13	31.16	0.54	6.06	1.65	5.04	2.68	7.46	1.09	10.45	1.48	6.46	1.17	12.45
OPEN-VINS	1.76	10.45	0.91	17.12	1.18	12.63	0.94	25.58	1.04	41.31	0.71	6.56	1.66	6.86	2.87	7.46	0.94	9.70	1.40	5.44	1.25	14.64
OKVIS	2.60	17.12	1.00	15.65	1.08	19.39	1.10	24.48	0.95	32.52	0.88	9.19	2.27	10.40	2.02	10.45	2.19	13.43	1.90	11.51	1.50	15.31
VINS-Mono	0.78	16.20	0.86	12.18	1.82	18.25	2.51	31.41	0.94	24.29	0.82	7.27	2.74	10.21	5.15	15.25	2.13	9.29	2.57	12.43	3.43	22.29
VIREO-35cm	1.98	10.23	0.78	14.45	0.78	11.05	0.96	24.43	1.06	29.16	0.55	6.06	1.66	5.08	2.66	7.49	1.08	10.45	1.44	6.31	1.44	15.45
VIREO-25cm	1.67	7.82	0.54	6.68	0.81	6.61	0.98	11.44	0.75	5.91	0.53	4.93	1.41	3.50	1.99	8.32	1.08	4.92	1.44	6.11	0.95	9.31
VIREO-15cm	1.38	6.18	0.39	3.68	0.48	3.61	0.85	7.44	0.74	5.72	0.42	4.81	1.11	3.51	1.59	3.72	0.75	4.22	1.24	2.13	0.85	5.71
VIREO-5cm	1.35	6.16	0.33	3.51	0.47	3.64	0.81	7.45	0.75	5.72	0.44	4.61	1.09	3.53	1.60	3.72	0.76	4.23	1.24	2.11	0.84	5.70

Table 2: Trajectory Accuracy (Translation(cm)/Rotation(degree)) on the RPNG AR Table Dataset. Top three results in translation and rotation are evaluated independently and highlighted in **Red** (1st), **Blue** (2nd), and **Green** (3rd).

Method	table_01	table_02	table_03	table_04	table_05	table_06	table_07	table_08								
RGB Modality																
Splat-SLAM	11.45	25.64	13.55	46.02	17.65	32.58	1.45	8.14	2.55	7.86	1.35	4.99	1.58	6.15	<i>1.45</i>	<i>8.31</i>
TTT3R	12.45	23.47	9.15	38.24	13.22	29.31	1.36	<i>5.49</i>	55.15	71.05	14.20	25.13	12.35	28.15	10.61	18.06
CUT3R	13.56	41.24	14.42	42.60	24.26	45.34	2.30	6.45	62.20	88.45	17.15	44.70	17.75	34.90	15.52	22.20
HI-SLAM2	0.44	2.66	1.02	2.41	1.23	3.51	1.06	4.86	F	0.52	2.55	0.65	2.32	1.12	5.49	
RGB + IMU Input																
$\sqrt{VINS}$	0.52	6.35	<i>0.91</i>	<i>2.45</i>	<i>1.37</i>	<i>5.54</i>	0.87	7.13	0.76	4.92	0.67	4.49	<i>0.85</i>	<i>4.65</i>	1.98	12.44
OPEN-VINS	0.45	6.83	0.85	2.74	1.44	5.68	0.83	7.54	0.78	<i>5.07</i>	0.68	4.90	0.94	<i>4.81</i>	2.02	12.91
OKVIS	2.48	9.40	2.01	7.17	3.94	15.43	2.05	16.42	<i>0.77</i>	24.35	0.74	10.26	2.07	13.48	1.54	18.18
VINS-Mono	1.62	<i>5.48</i>	1.32	3.40	1.47	11.64	1.75	7.26	1.12	7.45	0.98	5.53	1.67	9.36	5.03	23.34
Ours	<i>0.48</i>	2.83	0.87	2.19	1.22	3.60	0.80	2.48	0.74	4.07	0.64	4.46	0.84	3.08	0.92	3.70

test-time optimization—recovering Completion Ratios around 20% to 30%—it still suffers from severe structural distortion when compared to SLAM-based architectures. In contrast, our VIREO achieves competitive reconstruction quality, performing on par with HI-SLAM2 while substantially outperforming the purely feed-forward baselines. By strictly confining geometric consistency between adjacent pointmaps with EDF-optimizer, our system suppresses the inter-keyframe misalignment, yielding an accurate and complete metric 3D mesh.

Cross-Frame Geometric Consistency and Visualization To evaluate cross-frame geometrical consistency, CUT3R and TTT3R maps are metric-aligned via $Sim(3)$ transformations. We also introduce an ablation baseline (CUT3R + $\sqrt{VINS}$ Pose) by removing the EDF-based Optimizer while keeping the motion tracker and feed-forward 3D network. This strictly isolates and validates the absolute necessity of our EDF-based pose refinement. Specifically, we isolate the overlapping regions between adjacent keyframes and compute the point-to-point Euclidean distance between their reconstructed pointmaps. The resulting discrepancies are visualized as error heatmaps, capped at 2.0m, where larger values indicate larger geometric misalignment.

Table 3: Trajectory Accuracy (Translation(cm)/Rotation(degree)) on the FAST-LIVO2 Dataset. The top three results are highlighted in **Red** (1st), **Blue** (2nd), and **Green** (3rd). “F” denotes tracking failure.

Method	CBD1	CBD2	HKU	Retail	SYSU1
<i>RGB Input</i>					
Splat-SLAM	<i>3.45</i> / <i>5.32</i>	3.64 / 7.56	2.12 / 3.78	F	F
DROID-SLAM	14.45 / 64.25	7.95 / <i>15.82</i>	6.45 / 13.50	6.18 / 18.81	6.28 / 75.45
HI-SLAM2	2.12 / 4.35	<i>3.75</i> / 24.48	<i>2.75</i> / 14.41	2.81 / 7.79	2.31 / 12.45
TTT3R	7.72 / 14.35	6.45 / 20.56	5.48 / <i>11.35</i>	5.72 / <i>13.45</i>	<i>3.41</i> / <i>10.75</i>
CUT3R	34.35 / 634.02	15.45 / 538.15	28.85 / 345.12	23.45 / 651.00	35.15 / 741.56
<i>RGB + IMU Input</i>					
$\sqrt{VINS}$	6.96 / 13.75	4.75 / 18.19	3.12 / 16.45	<i>3.91</i> / <i>17.75</i>	4.05 / <i>12.19</i>
Ours	2.06 / 4.74	3.23 / 9.32	<i>2.82</i> / <i>12.23</i>	2.61 / <i>17.75</i>	2.04 / 9.46

Table 4: Dense 3D Reconstruction Quality on the RPNG AR Table Dataset. The top three results for each metric are highlighted in **Red** (1st), **Blue** (2nd), and **Green** (3rd). Note that for Accuracy and Completion, lower is better ($\downarrow$), whereas for Completion Ratio, higher is better ($\uparrow$).

Method	Metric	table_01	table_02	table_03	table_04	table_05	table_06	table_07	table_08
TTT3R	Acc.[cm] $\downarrow$	<i>28.53</i>	<i>23.93</i>	<i>13.40</i>	<i>25.49</i>	<i>23.45</i>	<i>24.02</i>	<i>33.34</i>	<i>23.03</i>
	Comp.[cm] $\downarrow$	<i>23.04</i>	<i>24.10</i>	<i>23.42</i>	<i>26.09</i>	<i>24.42</i>	<i>24.29</i>	<i>24.03</i>	<i>23.87</i>
	Comp.Rat[%] $\uparrow$	<i>22.75</i>	<i>29.61</i>	<i>22.10</i>	<i>31.49</i>	<i>29.84</i>	<i>28.91</i>	<i>27.01</i>	<i>25.38</i>
CUT3R	Acc.[cm] $\downarrow$	84.60	83.31	73.97	60.05	54.14	91.74	46.31	74.32
	Comp.[cm] $\downarrow$	75.56	84.48	66.90	66.31	68.46	90.16	51.40	75.01
	Comp.Rat[%] $\uparrow$	1.85	1.86	2.23	2.56	1.99	1.26	2.63	1.91
HI-SLAM2	Acc.[cm] $\downarrow$	4.84	9.07	12.05	7.98	5.06	5.32	3.34	6.92
	Comp.[cm] $\downarrow$	8.65	9.55	14.64	9.39	8.43	7.54	4.57	6.78
	Comp.Rat[%] $\uparrow$	81.96	75.78	64.50	71.82	71.07	82.09	91.41	81.04
Ours	Acc.[cm] $\downarrow$	7.16	7.74	7.26	7.85	4.60	8.72	3.02	6.13
	Comp.[cm] $\downarrow$	8.23	8.08	8.11	6.15	5.84	8.52	4.56	6.62
	Comp.Rat[%] $\uparrow$	79.29	81.21	77.80	82.95	82.64	72.10	94.04	89.88

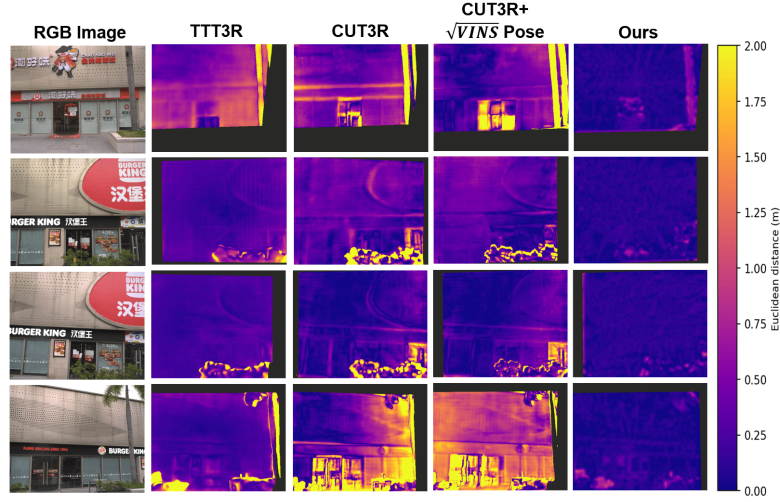


Fig. 5: Point-to-point Euclidean distance heatmap of overlapping regions between adjacent keyframes. Maximum error is capped at 2.0 meters. While baseline networks show significant spatial discrepancies (yellow/orange regions), our method maintains strict geometric consistency across frames.

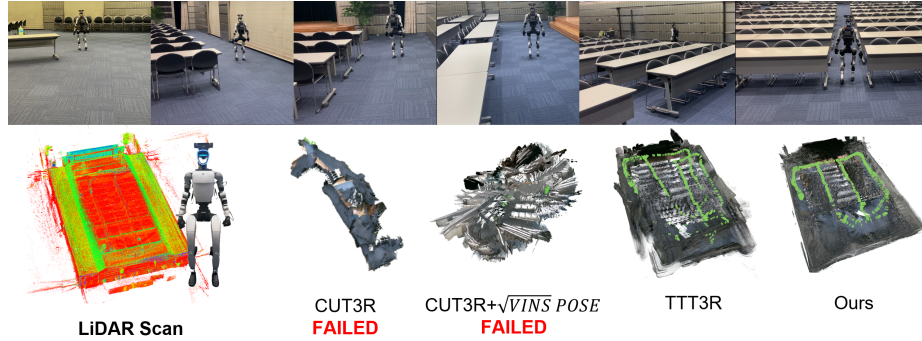


Fig. 6: Real-world deployment on the Unitree G1 humanoid robot in a challenging conference hall (highly repetitive textures and structures). While feed-forward baselines (processed offline on an A100 GPU) suffer from catastrophic map collapse or severe geometric drift, our VIREO (processed fully online on a Jetson Orin NX) successfully maintains a straight, drift-free trajectory and recovers a dense mesh that structurally aligns with the LiDAR reference scan.

As shown in Fig. 5, both CUT3R and TTT3R exhibit significant spatial misalignment within overlapping regions, while our method produces consistently low-error maps. These results indicate that the covariance-adaptive voxel distance field effectively enforces cross-frame geometric consistency. The whole-map visualization in Fig. 3 and the intermediate reconstructions in Fig. 4 show that, while the baseline networks suffer from severe structural bending and ghosting artifacts, ours produces globally consistent, crisp, and drift-suppressed dense reconstructions across diverse environments.

6.2 Unitree G1 Embodied Deployment

To demonstrate the real-world robotic deployment, we evaluate the proposed VIREO on large-scale sequences collected by a Unitree G1 humanoid robot as shown in Fig. 6. The data was processed in real-time on an NVIDIA Jetson Orin NX (16GB), identical to the actual onboard computing platform. The robot navigated through a large-scale conference hall, recording a continuous sequence of over 5900 frames, yielding roughly 1700 keyframes. Note that this environment presents challenges for visual tracking and reconstruction due to highly repetitive carpet textures, uniform lighting, and identical rows of desks and chairs.

System Efficiency and Scalability As shown in Fig. 7, feed-forward networks (CUT3R, TTT3R) face critical memory bottlenecks due to simultaneous frame encoding, triggering OOM crashes before 600 frames on Jetson Orin NX. Conversely, VIREO processes keyframes sequentially, eliminating expanding buffers. Varying voxel resolutions (5cm to 35cm) only slightly shift the runtime—from 7.8 to 9.8 FPS—without sacrificing stability, since EDF-based voxel-map construction is a one-time, highly-parallel operation. This allows VIREO to scale

to $> 1,000$ keyframes at ~ 9.1 FPS (VIREO-15cm) while maintaining a near-constant memory footprint, outperforming the rapidly degrading baselines.

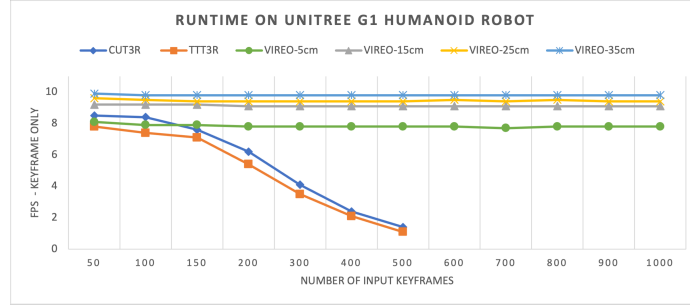


Fig. 7: Keyframe runtime (FPS) against varying number of input keyframes. While baseline networks exhibit drastic performance degradation and eventually trigger Out-of-Memory (OOM) failures, our system maintains a stable, constant-time processing speed regardless of the trajectory length and voxel resolution.

Online Large-scale Reconstruction Quality Due to the memory constraints demonstrated in Sec. 6.2, CUT3R and TTT3R required offline processing on a server-grade A100 (40GB) GPU for the humanoid sequence. Despite unlimited resources, Fig. 6 shows they still fail catastrophically: CUT3R yields an unrecognizable twisted mesh (even with external VINS guidance), while TTT3R suffers from trajectory drift, introducing severe discontinuities along the straight aisles. VIREO, running *online* on the Orin NX, successfully mitigates visual ambiguities in repetitive textures, recovering a globally consistent, straight 3D point cloud that structurally aligns with the LiDAR reference.

6.3 Ablation on EDF-based Pose Optimization

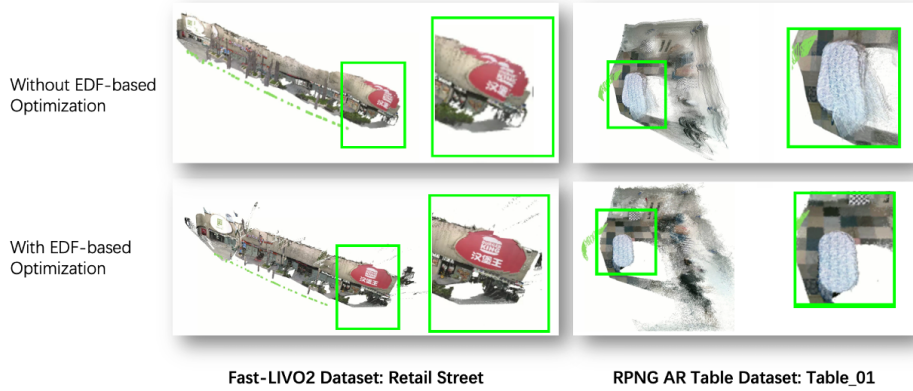
To isolate the contribution of EDF-based pose optimization, we compare full VIREO with CUT3R+ $\sqrt{\text{VINS}}$ Pose, which retains the same dense reconstruction pipeline but disables EDF refinement. On the eight RPNG AR Table sequences, EDF reduces the average translational/rotational ATE from 0.991 cm/5.996° to 0.814 cm/3.301°, reduces Accuracy/Completion errors from 19.26/18.93 cm to 6.56/7.01 cm, and increases Completion Ratio from 52.65% to 82.49%, as shown in Table 5. The qualitative results in Fig. 8 further show that disabling EDF leads to cross-frame ghosting and blurred structures, whereas EDF refinement produces sharper geometric boundaries and more consistent reconstruction. Together, these results demonstrate the effectiveness of EDF for improving both pose accuracy and cross-frame geometric consistency.

7 Conclusion and Future Work

While feed-forward 3D models offer strong local geometric priors, they inherently lack global metric consistency and online scalability. To address this issue, we

Table 5: Ablation of EDF-based pose optimization on the RPNG AR Table dataset. Lower is better for Accuracy and Completion. Higher is better for Completion Ratio.

Method	Metric	table_01	table_02	table_03	table_04	table_05	table_06	table_07	table_08
CUT3R+ $\sqrt{\text{VINS}}$ Pose	Acc.[cm] $\downarrow$	21.12	14.65	18.23	22.38	19.36	13.42	18.11	26.82
	Comp.[cm] $\downarrow$	27.64	15.40	16.56	19.54	16.17	15.93	16.51	23.70
	Comp.Rat.[%] $\uparrow$	44.41	61.23	57.26	47.79	65.62	64.37	51.01	29.50
Ours	Acc.[cm] $\downarrow$	7.16	7.74	7.26	7.85	4.60	8.72	3.02	6.13
	Comp.[cm] $\downarrow$	8.23	8.08	8.11	6.15	5.84	8.52	4.56	6.62
	Comp.Rat.[%] $\uparrow$	79.29	81.21	77.80	82.95	82.64	72.10	94.04	89.88

**Fig. 8:** Qualitative ablation on the EDF-based pose optimizer. The top row exhibits reconstruction artifacts, such as ghosting and blurred edges on the billboard and desk, when the optimizer is disabled. The bottom row demonstrates that our EDF-based refinement effectively restores multi-view consistency, yielding sharp, continuous geometric boundaries and significantly higher reconstruction quality.

have developed the online visual-inertial-based reconstruction with Euclidean-distance optimization, termed VIREO. We first integrated inertial measurements and a visual-inertial motion tracker into metric-scale feed-forward 3D reconstruction, and designed a novel EDF-based pose optimizer that refines both motion estimates and the global dense map by minimizing EDF-based residuals. Instead of treating reconstruction and tracking as isolated pipelines, our VIREO exploits a bidirectional interaction: sparse VINS features anchor the unscaled neural pointmaps to a metric scale, while structured geometric constraints extracted from dense pointmaps refine pose estimates via a fast EDF-based pose optimization. Extensive experimental results have demonstrated the effectiveness of the proposed VIREO system. In the future, we will extend to dynamic environments and 3D scene understanding to further empower high-level robotic autonomy.

Acknowledgments This work was partially supported by University of Delaware College of Engineering, NASA/EPSCoR Seed Grant, NSF IIS-2410019, and Toyota Motor Engineering & Manufacturing North America.

References

1. Bloesch, M., Burri, M., Omari, S., Hutter, M., Siegwart, R.: Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research* **36**(10), 1053–1072 (2017)
2. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Sparse iterative closest point. *ACM Transactions on Graphics (TOG)* **32**(6), 1–11 (2013)
3. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Transactions on Robotics* **37**(6), 1874–1890 (2021)
4. Chen, C., Geneva, P., Peng, Y., Lee, W., Huang, G.: Monocular visual-inertial odometry with planar regularities. In: *Proc. of the IEEE International Conference on Robotics and Automation*. London, UK, (2023)
5. Chen, C., Geneva, P., Peng, Y., Lee, W., Huang, G.: Optimization-based vins: Consistency, marginalization, and fe. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE (2023)
6. Chen, C., Peng, Y., Huang, G.: Visual-inertial state estimation with decoupled error and state representations. In: *International Workshop on the Algorithmic Foundations of Robotics* (2024)
7. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645* (2025)
8. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: Fast semi-direct monocular visual odometry. In: *2014 IEEE international conference on robotics and automation (ICRA)*. pp. 15–22. IEEE (2014)
9. Geneva, P., Eickenhoff, K., Lee, W., Yang, Y., Huang, G.: OpenVINS: a research platform for visual-inertial estimation. In: *Proc. of the IEEE International Conference on Robotics and Automation*. Paris, France (2020), https://github.com/rpng/open_vins
10. Grompone von Gioi, R., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(4), 722–732 (2010). <https://doi.org/10.1109/TPAMI.2008.300>
11. Huang, G.: Visual-inertial navigation: A concise review. In: *Proc. International Conference on Robotics and Automation*. Montreal, Canada (May 2019)
12. Kulkarni, A.M., Poulakakis, I., Huang, G.: Learning neural observer-predictor models for limb-level sampling-based locomotion planning. *arXiv preprint arXiv:2510.22789* (2025)
13. Leroy, V., Weinzaepfel, P., Brégier, R., Comport, H., Obozinski, G., Cabon, Y., Revaud, J.: MAST3R: Grounding image matching in 3d. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024)
14. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research* **34**(3), 314–334 (2015)
15. Magnusson, M., Lilienthal, A., Duckett, T.: Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics* **24**(10), 803–827 (2007)
16. Peng, Y., Chen, C., Huang, G.: Ultrafast square-root filter-based vins. In: *Proc. International Conference on Robotics and Automation*. Yokohama, Japan (May 2024)
17. Peng, Y., Chen, C., Wu, K., Huang, G.: $\sqrt{\text{vins}}$: Robust and ultrafast square-root filter-based 3d motion tracking. *IEEE Transactions on Robotics* **41**, 6570–6589 (2025). <https://doi.org/10.1109/TRO.2025.3626607>

18. Qin, T., Li, P., Shen, S.: VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics* **34**(2), 1004–1020 (2018)
19. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International Conference on Computer Vision. pp. 2564–2571 (2011). <https://doi.org/10.1109/ICCV.2011.6126544>
20. Sandström, E., Zhang, G., Tateno, K., Oechsle, M., Niemeyer, M., Zhang, Y., Patel, M., Van Gool, L., Oswald, M., Tombari, F.: Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 1680–1691 (2025)
21. Segal, A., Hähnel, D., Thrun, S.: Generalized-icp. In: *Robotics: Science and Systems*. vol. 2, p. 435. Seattle, WA (2009)
22. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The Replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
23. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 6229–6238 (2021)
24. Teed, Z., Deng, J.: DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. *Advances in neural information processing systems* **34**, 7166–7177 (2021)
25. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025)
26. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 10510–10522 (2025)
27. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUST3R: 3d reconstruction of the wild without known camera poses. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 8622–8632 (2024)
28. Zhang, W., Cheng, Q., Skuddis, D., Zeller, N., Cremers, D., Haala, N.: Hi-slam2: Geometry-aware gaussian slam for fast monocular scene reconstruction. *IEEE Transactions on Robotics* **41**, 6478–6493 (2025)
29. Zheng, C., Xu, W., Zou, Z., Hua, T., Yuan, C., He, D., Zhou, B., Liu, Z., Lin, J., Zhu, F., Ren, Y., Wang, R., Meng, F., Zhang, F.: Fast-livo2: Fast, direct lidar-inertial-visual odometry. *IEEE Transactions on Robotics* **41**, 326–346 (2025). <https://doi.org/10.1109/TR0.2024.3502198>
30. Zhuo, D., Zheng, W., Guo, J., Wu, Y., Zhou, J., Lu, J.: Streaming 4d visual geometry transformer. arXiv preprint arXiv:2507.11539 (2025)

A Analytic Jacobian Derivations in EDF-based Pose Optimization

To solve the non-linear least-squares problem using the Levenberg-Marquardt algorithm, we derive the analytic Jacobian matrix $\mathbf{J}_i$ for each sample point. By applying a small left-perturbation $\delta\boldsymbol{\xi} \in \mathfrak{se}(3)$ to the pose $\mathbf{T}$, the Jacobian is computed using the chain rule:

$$\mathbf{J}_i = \frac{\partial r_i^{EDF}(\exp(\delta\boldsymbol{\xi}^\wedge)\mathbf{T})}{\partial \delta\boldsymbol{\xi}} = \frac{\partial r_i^{EDF}(\exp(\delta\boldsymbol{\xi}^\wedge)\mathbf{T})}{\partial ({}^W\mathbf{p}_i)} \frac{\partial ({}^W\mathbf{p}_i)}{\partial \delta\boldsymbol{\xi}} \quad (5)$$

where the residual, $r_i^{EDF}(\mathbf{T})$, is defined in (3). The gradient of EDF $\nabla\mathcal{D}$, is approximated by computing differences to EDF values of neighbors:

$$\frac{\partial r_i^{EDF}(\exp(\delta\boldsymbol{\xi}^\wedge)\mathbf{T})}{\partial ({}^W\mathbf{p}_i)} \approx \nabla\mathcal{D}({}^W\mathbf{p}_i) \quad (6)$$

where $\nabla\mathcal{D}({}^W\mathbf{p}_i)$ is the discrete spatial gradient evaluated at the voxel cell $\mathbf{v}$. The second term is the standard point perturbation Jacobian with respect to $SE(3)$:

$$\frac{\partial ({}^W\mathbf{p}_i)}{\partial \delta\boldsymbol{\xi}} = [\mathbf{I}_{3 \times 3} \quad -[{}^W\mathbf{p}_i]_\times] \quad (7)$$

Combining these components, the complete analytic Jacobian is given by:

$$\mathbf{J}_i = \nabla\mathcal{D}({}^W\mathbf{p}_i)^\top [\mathbf{I}_{3 \times 3} \quad -[{}^W\mathbf{p}_i]_\times] \quad (8)$$

B Analytic SRF Information Matrix Update

The refined pose $\mathbf{T}$ is fed back to VINS as a global measurement. With covariance $\mathbf{R} = (\mathbf{J}^\top \mathbf{J})^{-1}$, where $\mathbf{J}$ is the optimization Jacobian from Levenberg–Marquardt optimizer in Sec.5.3 and measurement Jacobian $\mathbf{H} = \mathbf{I}_{6 \times 6}$, the standard information matrix update is:

$$\mathbf{U}_{refine}^T \mathbf{U}_{refine} = \mathbf{U}_{pose}^T \mathbf{U}_{pose} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \quad (9)$$

where $\mathbf{U}_{pose}$ is the pose-related 6×6 upper-triangular block extracted from the $\sqrt{\text{VINS}}$ [17]’s full information matrix $\mathbf{U}$. We define a projection matrix $\mathbf{A} = \mathbf{R}^{-1/2} \mathbf{H} \mathbf{U}_{pose}^{-1}$ to factor the update:

$$\mathbf{U}_{refine}^T \mathbf{U}_{refine} = \mathbf{U}_{pose}^T (\mathbf{I} + \mathbf{A}^T \mathbf{A}) \mathbf{U}_{pose} \quad (10)$$

Applying a Cholesky decomposition $\mathbf{F}^T \mathbf{F} = \mathbf{I} + \mathbf{A}^T \mathbf{A}$ yields an upper-triangular factor $\mathbf{F}$, simplifying the SRF update to an ultra-fast local transformation:

$$\mathbf{U}_{refine} = \mathbf{F} \mathbf{U}_{pose} \quad (11)$$

After this localized update, we replace the old $\mathbf{U}_{pose}$ block in the full matrix $\mathbf{U}$ with $\mathbf{U}_{refine}$ to obtain the new updated full upper-triangular information matrix, $\mathbf{U}_{update}$.

$$\mathbf{U}_{update}^T \mathbf{y} = \mathbf{b}, \quad \mathbf{U}_{update} \delta \mathbf{x} = \mathbf{y} \quad (12)$$

where $\mathbf{b}$ is the zero-padded full information vector, whose pose-associated 6-dimensional block is $\mathbf{R}^{-1} \mathbf{r}$, with $\mathbf{r} = \log(\mathbf{T}_{VINS}^{-1} \mathbf{T})^\vee \in \mathbb{R}^6$. The computed update $\delta \mathbf{x}$ is then applied to system state.