VGGT-SLAM++

Avilasha Mandal[1] Rajesh Kumar[2] Sudarshan Sunil Harithas[3] Chetan Arora[1]
[1]Indian Institute of Technology Delhi [2]Addverb Technologies [3]Brown University

cs1221631@iitd.ac.in

Figure 1. VGGT-SLAM++ provides an end-to-end SLAM architecture that stabilizes transformer-based odometry by using a low-fidelity geometric representation to support a high-cadence optimization back-end. The trajectories in the upper row are the odometry based trajectories while the lower row corresponds to the corrected trajectories when stabilised by our back-end.

Abstract

*We introduce **VGGT-SLAM++**, a complete visual SLAM system that leverages the geometry-rich outputs of the Visual Geometry Grounded Transformer (VGGT). The system comprises a visual odometry (front-end) fusing the VGGT feed-forward transformer and a Sim(3) solution, a Digital Elevation Map (DEM)-based graph construction module, and a back-end that jointly enable accurate large-scale mapping with bounded memory. While prior transformer-based SLAM pipelines such as VGGT-SLAM rely primarily on sparse loop closures or global Sim(3) manifold constraints—allowing short-horizon pose drift—VGGT-SLAM++ restores high-cadence local bundle adjustment (LBA) through a spatially corrective back-end. For each VGGT submap, we construct a dense planar-canonical DEM, partition it into patches, and compute their DINOv2 embeddings to integrate the submap into a covisibility graph. Spatial neighbors are retrieved using a Visual Place Recognition (VPR) module within the covisibility window, triggering frequent local optimization that stabilizes trajectories. Across standard SLAM benchmarks, VGGT-SLAM++ achieves state-of-the-art accuracy, substantially reducing short-term drift, accelerating graph convergence, and maintaining global consistency with compact DEM tiles and sublinear retrieval.*

1. Introduction

Cameras and LiDARs have been widely used for robot and autonomous vehicle localization in 3D environments for several decades [51, 62, 67, 74]. The camera pose estimation part of the SLAM system consists of two main components: (i) a high-frequency odometry (front-end) that provides relative pose estimates, and (ii) a slower

back-end that optimizes the spatial relationships among map entities and camera poses. Classical visual odometry pipelines rely on optical-flow-based feature tracking [22, 35]] or feature-descriptor-based matching [3, 33, 45], whose robustness is strongly affected by the repeatability of feature detection and the reliability of feature matching [30, 46, 47, 56]. Recent trends emphasize the use of semantically informed [37, 58] or geometrically informed [60, 64] features and learned matchers that improve data association under challenging conditions [11, 46]. Transformer-based architectures in computer vision has inspired a new family of SLAM and odometry systems that leverage transformer models for feature extraction, matching, and global reasoning [12, 29, 66, 75].

Modern RGB SLAM systems [15, 75] based on dense transformers [66] (e.g., VGGT-SLAM [36]) have made impressive progress in long-range consistency through robust global loop closure. However, their corrective behavior between loop events often remains coarse: the front-end [42, 47] accumulates drift, and the back-end [8, 57] typically waits for large-baseline revisits to activate strong constraints.

We address this gap with a **spatially corrective back-end** that prioritizes **high-frequency local bundle adjustment (LBA)** [6, 38, 39, 55] over exclusive reliance on infrequent global bundle adjustment (loop closures). Our key observation is that short temporal windows and neighboring submaps provide enough multi-view geometry to curb drift if they can be identified and optimized quickly. We therefore design a pipeline that (i) raises the cadence of local corrections in the back-end to keep pace with the transformer front-end, and (ii) feeds these corrections with compact, structure-preserving map evidence to make each local optimization both cheaper and more discriminative. To this end, we introduce **DEM-augmented submaps** [21] and a **covisibility search** [39, 44]. Each submap exports a dense **Digital Elevation Map (DEM)**—a compact, spatially coherent projection that preserves local affine structure with enough coherent features for networks like DINOv2 [43, 49, 71] to generate structurally aware embeddings. DEMs (see Fig. 2) a lightweight representation for geometry-aware alignment passes through encoders for generating candidate embeddings for retrieval. We synthesize a covisibility graph to reduce search space and time for loop detection as we leverage, AnyLoc [25] for Visual Place Recognition (VPR) within a relevant covisibilty window avoiding redundant search throughout the entire map space. We show that **loop detection** [18, 39] via VPR [25, 34] performed directly in the DEM domain produces sufficiently informative constraints to **synthesize the spatial hierarchy**. Although DEMs provide a compact representation of large scale maps, they preserve local affine structure. It is observed that DEMs support structural matching, we can register a new submap using a VPR module (e.g. AnyLoc [25]). Spatial hierarchy is generated within the covisibility zone of neighboring submaps. The relative pose within the spatially connected submaps are further passed through a Sim(3) optimizer [53] to improve the pose of the injected submap.

RGB images have a low field-of-view (FOV) and retrieval methods that rely on per-frame features are susceptible to false positive detections. Submap-level DEMs have much larger FOV and provide rich features. VGGT-SLAM++ not only achieves high-cadence spatial correction of front-end drifts but also overcomes one of the key limitations of the VGGT-SLAM—in planar scenes. By leveraging a **Digital Elevation Map (DEM)**-based loop detection, our approach anchors local submaps in a geometry-preserving representation that complements transformer-derived semantics, fetching the best out of both worlds. Hence Digital Elevation Maps provide large-scale structural coherence for covisibility region discovery in long sequences (see Fig. 1), and precise geometric cues for local alignment and loop detection in planar environments (visualisation of trajectory in planar scene, TUM RGB-D frieburg1 floor scene is at (Fig. 4). Consequently, our framework unifies semantic scalability and geometric stability scenario across both long-horizon and near-planar trajectories. The major contributions are:

- Compact, geometry-preserving **DEM-augmented map representation** that remain compatible with encoders like DINOv2, enabling rich structural verification and retrieval.
- A **covisibility graph synthesis** via a local affine structure based search. The covisibility graph **reduces search space and complexity**, yielding faster but accurate loop detections leveraging a VPR module for candidate submap insertion within its covisibility region.
- A back-end that schedules **local bundle adjustment** at a higher cadence, curbing odometry drift between loop events leading to **high-frequency spatial correction.**

We validate the design on standard driving and robotics sequences, showing the presented method reduces short-term drift, accelerates graph stabilization, and maintains or improves final global consistency—while keeping runtime within practical budgets owing to the DEM-based cascaded search.

2. Related Work

Feed-Forward 3D Reconstruction Networks. Feed-forward transformers have recently emerged as a unifying paradigm for multi-view 3D reconstruction, replacing iterative structure-from-motion with direct geometric inference. **DUSt3R** [69] pioneered dense point and pose prediction from two uncalibrated images, fol-

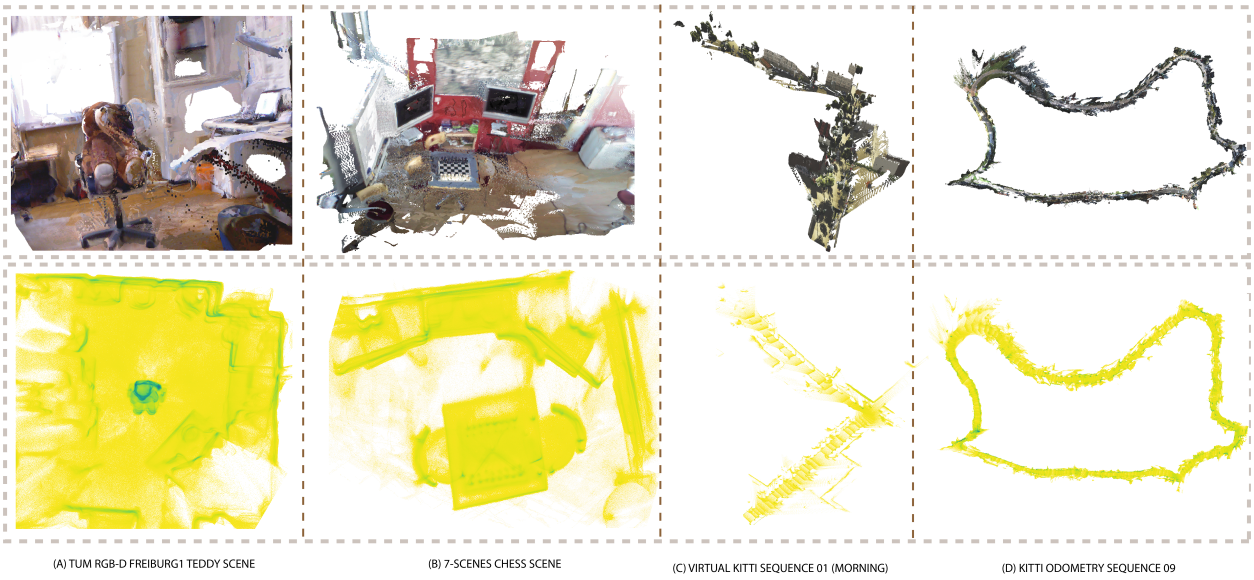

Figure 2. (A) DEM-based scene representation on the TUM RGB-D freiburg1 teddy dataset. The DEMs provide a compact 2.5D encoding retaining geometric structure. (B) DEM visualizations from the 7-Scenes dataset. (C) DEMs generated for the Virtual KITTI (Sequence 01) dataset. (D) A full KITTI Odometry (Sequence 09) sequence demonstrating a complete loop, illustrating the ability of DEMs and our SLAM back-end pipeline to maintain global consistency.

lowed by **MASt3R** [40] and **MASt3R-SfM** [14], which extended it to multi-view settings with learned correspondences and global refinement. Temporal extensions such as **Spann3R** [65] and **Cut3R** [68] integrate memory or recurrence for longer sequences, while **Pow3R** [23] and **Splatt3R** [50] generalize the formulation to mixed cues and Gaussian-splat representations. The **MapAnything** [26] system unifies over a dozen reconstruction tasks—including multi-view stereo, SfM, registration, and depth completion—within a single feed-forward transformer that directly regresses metric 3D scene geometry. Building on this progression, the **Visual Geometry Grounded Transformer (VGGT)** [66] scales feed-forward reconstruction to hundreds of frames, jointly predicting cameras, depth, and dense tracks in one pass. In our work, VGGT serves as a feed-forward submap generator whose outputs provide dense geometry and camera priors for spatially corrective $\mathrm{Sim}(3)$ optimization. Following submap generation, VGGT-SLAM aligns adjacent submaps by estimating a relative transformation that resolves projective ambiguity between their respective reconstructions.

Digital Elevation Maps (DEMs) for Compact, Structure-Aware Geometry. DEM canonicalization [21] has recently emerged in LiDAR loop detection/closure [9] to expose strong planar and height priors, achieving large bandwidth savings and viewpoint robustness via roll/pitch normalization and top-down discretization. FinderNet [21] demonstrates that DEMs can enable both robust loop detection while remaining highly data-efficient and generalizable, and that DEMs are amenable to learned embeddings without heavy augmentation. Our use of DEMs is different in motive but similar in advantage. We generate image-like DEMs from dense RGB submaps (not LiDAR) as a compact 2.5D substrate that (i) preserves affine structure for geometric verification, (ii) passes cleanly through structure-aware encoders (e.g., DINOv2 [43]) for retrieval, and (iii) accelerates covisibility synthesis. This gives us compactness and geometric fidelity with structural discriminability, enabling fast and precise local corrections with reduced global complexity.

Unlike FinderNet [21], which trains a specialized network for DEM features and targets 5-DOF loop detection on dense LiDAR point clouds, VGGT-SLAM++ uses DINOv2 to embed DEMs, for transformer-generated point maps with unconstrained DOF and different noise characteristics than LiDAR. This enables robust loop detection within a complete transformer-based visual SLAM pipeline.

3. Method

Review. VGGT-SLAM [36] scales VGGT [66] to long video sequences into metrically meaningful submaps and optimizing their relative alignment. The pipeline incrementally selects keyframes from incoming RGB frames based on disparity—measured using Lucas–Kanade [35] flow—between the current and previous keyframes. When the disparity exceeds a threshold $\tau_{\text{disparity}}$ (we used 40m), the frame is added to the current submap's image

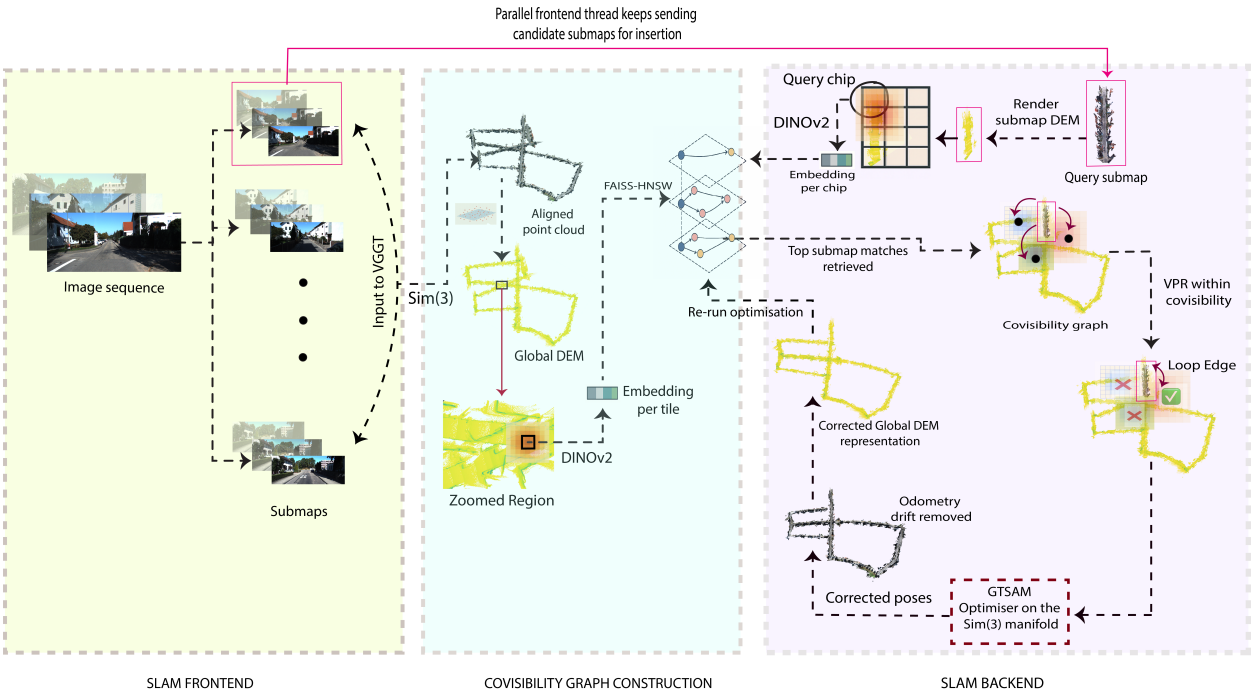

Figure 3. Complete pipeline overview. The proposed VGGT-SLAM++ system comprises three main components: (a) Front-end: A Sim(3) odometry module that optimizes the relative poses of submaps generated by the feed-forward VGGT network. (b) Covisibility graph construction: A DEM-based map representation is used to compute structure-aware embeddings leveraging DINOv2, and an averaged tile score is used to insert spatially consistent nodes and edges into the covisibility graph. (c) Back-end: An optimization module that organizes submaps into a spatial graph and performs optimization over the detected spatial constraints.

set I_{latest}. Once $|I_{\text{latest}}|$ reaches a fixed limit w (we used 32), it is finalized as a submap S_{latest}. To ensure temporal continuity, each submap inherits one transition (non-loop closure) frame M_{prior} from the previous submap, but unlike the original formulation that appended w_{loops} loop-closure frames, we assign $w_{\text{loops}} = 0$, to handle loop closure explicitly by our spatial drift-corrective back-end. Each submap is passed to VGGT, which reconstructs depth maps, cameras and point maps for the submap.

Overview. Our syst em, **VGGT-SLAM++** is an end to end SLAM framework using feed-forward transformer VGGT and a Sim(3) motion only solver. It augments the visual odometry with a spatially corrective solver over correlated submaps for continuous drift suppression. The overall pipeline (Fig. 3) operates as follows: sequential RGB frames are grouped into submaps based on frame disparity and processed by VGGT to yield camera poses. Each submap is aligned to its predecessor (temporal) through a $\mathrm{Sim}(3)$ transformation, forming a temporal consistent trajectory. We do a **depth thresholding** to remove floaters from cloud or sky based noise at the reconstruction horizon. This entire (temporal) point cloud alignment of all injected submaps formed from an initial robotic tele-operation is augmented to a global planar-canonical DEM map representation which is patched into smaller tiles at 2x2 meters maintaining robust resolution. Each DEM tile is embedded into a geometric feature space using DINOv2, producing compact descriptors indexed in a FAISS-HNSW [13] structure serving as scalable retrieval gallery, for a new query submap that arrives for registration, from the front-end tracking thread, which keeps running independently of the back-end, even after the tele-operation phase. For each such query submap, a **planar-canonical Digital Elevation Map (DEM)** is constructed in a similar fashion as the global DEM by patching into 2x2 meter chips serving as candidate queries for loop detection within a covisibility region and hence accurate registration of the submap. The front-end tracking thread continuously generates embeddings for these **query chips**—patched from candidate submaps awaiting insertion. These query embeddings of submap chips, are compared against the indexed global DEM tiles to identify spatially proximal submaps for covisibility reasoning [39, 44]. We build a covisibility window for the query submap, by comparing the DINOv2 descriptors of the 2x2 meters chips from the query submap's DEM and the 2x2 meter tiles from the global DEM. Within this covisibility window, we leverage AnyLoc [44] as a place recognition module, with the chips of submap awaiting insertion as the **queries** and the **tiles** in

its covisibility region of the global DEM as the retieval gallery for each one of those queries.

DEM-Augmented Submaps. For each submap, we convert the dense 3D points into a compact, geometry-preserving **Digital Elevation Map (DEM)** defined on a single globally consistent plane. For every point obtained from VGGT's dense reconstruction, we first robustly fit a global plane using RANSAC [16] and singular value decomposition [52]. We compute a bounding box in the pixel domain (a tile) and choose a resolution in meters-per-pixel. The continuous pixel coordinates are then discretized into a regular grid, and all heights falling into the same pixel are aggregated via a reducer (mean, max, or softmax-weighted average, we obtain the best results with the "softmax version"). This yields a tiled DEM representation where each tile stores a dense 2.5D height field [24, 41].

Structure-aware Embedding of DEM Tiles and Query Chips. Each global DEM tile τ_k is processed through a DINOv2 [43] encoder f_θ to obtain a feature vector with a weighted attention [2] over a 9x9 tile neighborhood considering τ_k sitting at the center of the arrangement.

$$v_k = \frac{\sum_j w_j \, m_j \, f_\theta(p_j)}{\sum_j w_j \, m_j}, \tag{1}$$

The DEM tile is first divided into small patches, producing a sequence of tokens $\{p_j\}$ that serve as the input to f_θ. The scalar w_j is a **Gaussian positional weight** that down-weights tokens near tile boundaries of the 9x9 neighborhood and emphasizes geometrically reliable central regions with respect to τ_k. The coefficient m_j is a **visibility mask** derived from the local gradient magnitude of the underlying DEM: flat or low-information regions receive lower weight, while edges, ramps, or height discontinuities contribute more strongly to the final descriptor. The resulting vector v_k is therefore a normalized, geometry-aware embedding that combines structure from DINOv2 with geometric salience from the DEM. Similar to the embedding mechanism of the global DEM tiles, every incoming query submaps are split into smaller **query chips** $\{\chi_q\}$ generated by the front-end tracking thread. These chips are passed through the same encoder f_θ with identical weighting logic (but the weighted embedding per chip is over the entire submap region patched to chips and not just a 9x9 neighborhood as in the global case) to produce query descriptors in the same embedding space. This dual embedding strategy enables new submap candidates to be compared directly against previously indexed global DEM tiles, forming the basis for fast, reliable covisibility discovery at scale.

FAISS-HNSW Covisibility Graph Construction. The embedded DEM tiles populate a global FAISS-HNSW [13] index that supports sublinear nearest-neighbor search across all previously constructed submaps. Let v_k denote the DINOv2 embedding of the k-th DEM tile, and let v_q denote the embedding of a query chip χ_q generated by the front-end tracking thread. For each query chip, we compute its similarity to every indexed DEM tile as

$$s(\chi_q, \tau_k) = \frac{v_q^\top v_k}{\|v_q\| \, \|v_k\|}, \tag{2}$$

where $v_q^\top v_k$ is the dot product between the two descriptors, and $\|v_q\|$, $\|v_k\|$ are their ℓ_2 norms. This normalized dot product corresponds to **cosine similarity** [70], which measures structural compatibility between the chip and tile embeddings while being invariant to descriptor magnitude.

For each query chip $\{\chi_q\}$, FAISS-HNSW returns a ranked list of approximate nearest-neighbor tiles τ_k with similarity scores $s(\chi_q, \tau_k)$. We then aggregate these matches at the *submap* level through simple voting: every retrieved tile contributes its raw similarity score to the score of its parent submap. Formally, the score for a submap $\mathcal{S}$ is

$$\text{Score}(\mathcal{S}) \mathrel{+}= \sum_{\tau_k \in \mathcal{S}} s(\chi_q, \tau_k), \tag{3}$$

where the sum ranges over all tiles belonging to $\mathcal{S}$. Submaps whose accumulated score exceeds a similarity threshold τ_s (or rank within the top-K hiararchies, we keep it to 10) are selected as **covisible neighbors** for the incoming candidate submap.

This process yields a sparse covisibility graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node in $\mathcal{V}$ corresponds to a submap, and each edge in $\mathcal{E}$ represents a strong structural relation discovered through the DEM–DINOv2 retrieval pipeline.

Visual Place Recognition. Once a set of top-K covisible submaps has been proposed by the FAISS–HNSW voting stage, we perform a loop detection within covisibility graph $\mathcal{G}$ using an AnyLoc-based [25] Visual Place Recognition (VPR) module. For each candidate submap pair $(i, j) \in \mathcal{E}$, AnyLoc retrieves a refined set of descriptor correspondences between the query chips and the tiles of the proposed neighbor submap. These chip-tile correspondences are once again aggregated at the *submap* level through simple voting: every retrieved tile contributes its AnyLoc retieval score to the score of its parent submap. We hence obtain candidate submap-to-submap loop edges spatially from our back-end.

Spatially Corrective Back-end Optimization. All submap-to-submap loop edges are passed to a spatially bounded back-end optimizer that operates concurrently with the front-end. Let $\{S_i\}$ denote the set of spatially connected submaps within the covisibility graph $\mathcal{G}$. For these submaps, the back-end seeks globally consistent similarity poses $\mathbf{T}_i \in \text{Sim}(3)$ by minimizing the

Figure 4. Preservation of geometric cues in the DEMs, alongside semantics help in accurate trajectory estimation in planar scenes like the TUM RGB-D foor scene as shown in the figure.

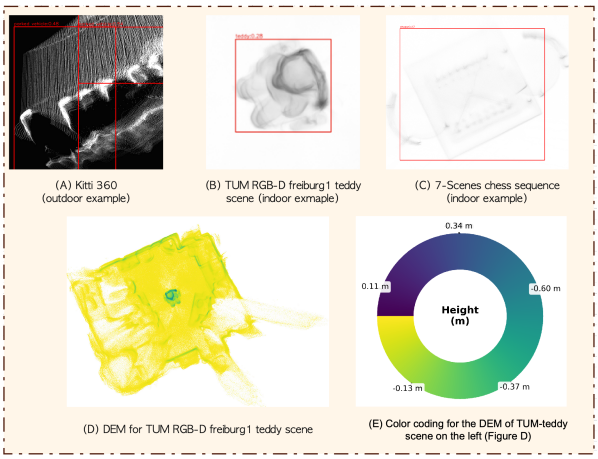

Figure 5. (A), (B), (C): zero-shot object detection from DEMs proving structure preservation. (D) DEM of TUM-teddy and (E) color coding.

weighted geodesic error [1] of all loop edges:

$$\min_{\{\mathbf{T}_i \in \mathrm{Sim}(3)\}} \sum_{(i,j)\in\mathcal{E}} \left\| \log_{\mathrm{Sim}(3)} \left(\mathbf{T}_j^{-1}\, \mathbf{T}_i\, \hat{\mathbf{T}}_{ij} \right) \right\|^2_{\Sigma_{ij}} . \quad (4)$$

Here, $\hat{\mathbf{T}}_{ij}$ is the estimated $\mathrm{Sim}(3)$ relative transform between the submaps acting as loop edges, and Σ_{ij} captures the per-edge uncertainty derived from descriptor consistency and 3D alignment residuals. The log-map $\log_{\mathrm{Sim}(3)}(\cdot)$ converts similarity transformations [28] into their tangent-space residuals [63], enabling standard Gauss–Newton optimization [4]. This optimization is invoked at a high cadence and acts as a **spatially corrective layer** over the front-end: it stabilizes trajectories, suppresses drift between loop events, and maintains global consistency. Together with DEM-based geometry and transformer-derived priors, our system yields a compact, scalable SLAM system capable of robust long-horizon operation.

4. Experiments

Datasets and Setup. We evaluate VGGT-SLAM++ on a diverse set of datasets encompassing both synthetic and real-world conditions: KITTI Odometry [20], TUM RGB-D [54], 7-SCENES [48], Virtual KITTI [17], EuRoC MAV [5] (shown in Appendix A1). Visualisations of corrected trajectories in some sequences from the above dataset are shown in Fig. 6. All experiments are conducted on an NVIDIA RTX 4090 GPU with 24 GB VRAM and AMD Ryzen Threadripper PRO 5955WX 16-Cores CPU with 32GB RAM. DEM rendering, DINO embedding, and local bundle adjustment (LBA) run on GPU; FAISS-HNSW indexing executes on CPU, ensuring constant memory usage per submap. The results are benchmarked using the root mean squared Absolute Trajectory Error (ATE) [73] (ATE rmse) in meters.

Memory Profile. At inference time, only the current submaps' (in covisibility window) VGGT features, their DEM raster, and the DINOv2 patch tokens for respective tiles from retrieval gallery or query chips reside in GPU memory [submap point clouds outside the covisibility window reside in disk], keeping VRAM usage well within the budget of an RTX 4090 (typically 20 GB during full operation). All global DEM tiles are stored on the CPU as compressed 2.5D grids (approximately 1–1.2 MB each); within the 32 GB system RAM available on the RTX 4090 server. The FAISS–HNSW index resides fully on CPU, growing sublinearly in memory due to hierarchical graph compression and fixed 768-dimensional tile descriptors. The VGGT-SLAM++ front- end runs at ∼16 FPS and it's spatially corrective back-end runs at 1.89 FPS, with bounded memory usage (∼8 GB RAM, ∼20 GB VRAM), averaged across datasets referred in Tables 1, 4, 3, and 2, showing bounded memory compared to prior work like DROID-SLAM [61] with 8GB front-end and 24GB back-end.

Structure-aware DEM. As shown in Fig. 5 (A, B, C) zero shot object detection by GDINO [32] ran on DEMs gives accurate detections for a 'parked vehicle' (Kitti 360 scene), 'teddy' (from TUM RGB-D freiburg1 scene) and 'chess's (7-scenes). These prove DEMs as powerful scene augmentations preserving rich structural cues. The colored version of DEMs, shown in Figure. 5(D, E) is only used for visualisations by humans whereas DINOv2 interpretes grayscale version. Yellow is the ground plane, darker the shade of green higher the height of the real world point.

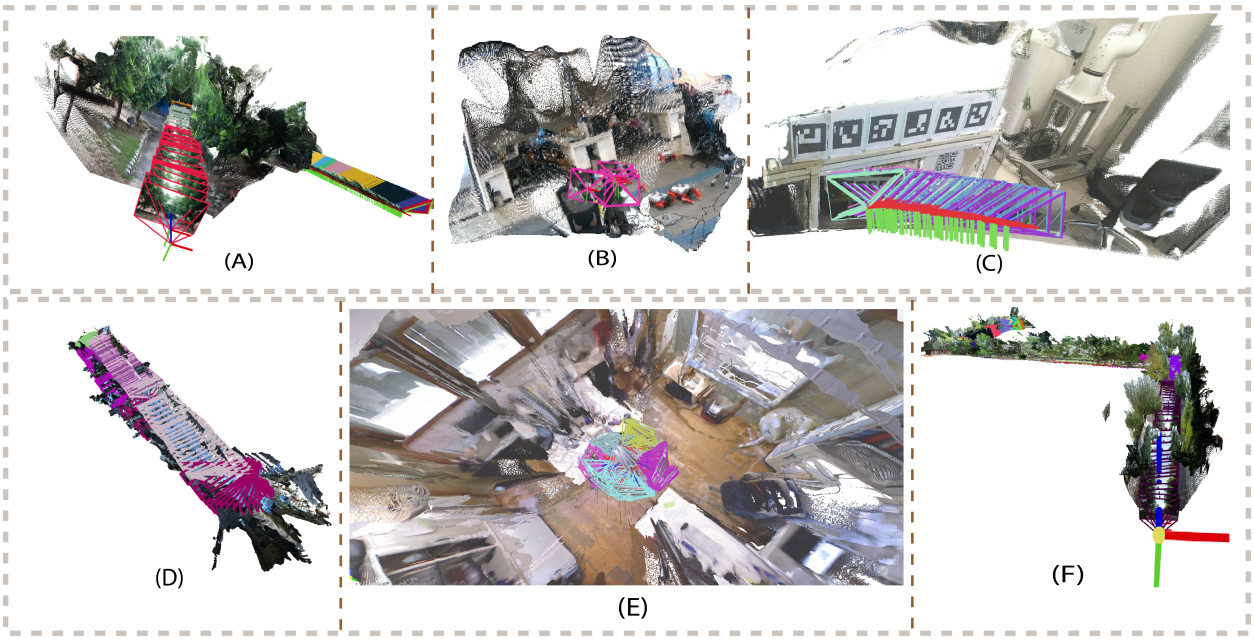

Figure 6. VGGT-SLAM++ results for : (A) custom data (406.8m) recorded by GoPro HERO10 camera with GPS groundtruth with 2m precision. (ATE RMSE 18 ± 2 m); (B) custom data (1.8m) recorded by a OAK-1 camera with a Humanoid robot kinematics groundtruth (ATE RMSE 0.02m); (C) custom data (1.8m) recorded by a OAK-1 camera with Cobot forward kinematics groundtruth (ATE RMSE 0.01m); (D) KITTI Odometry 06 sequence (1230 m; ATE RMSE 13.65m); (E)TUM RGB-D 360 scene (5.82m; ATE RMSE 0.042m); (F) custom data (287.381m) path recorded by the GoPro HERO10 camera with GPS groundtruth with 2m precision. (ATE RMSE 7.17 ± 2 m).

4.1. Quantitative Results

Experimental results are summarized in Tables 1–4. As shown, **VGGT-SLAM++** achieves comparable or superior results performance across all RGB datasets with uncalibrated camera inputs, with **DROID-SLAM**, **MASt3R-SLAM**, and **VGGT-SLAM**. Notably, VGGT-SLAM++ is among the first transformer-based architectures using uncalibrated camera sources to achieve ATE comparable to calibrated-camera systems such as MASt3R-SLAM. Results on the Virtual KITTI benchmark (Table 4) also indicate strong robustness to varying weather and illumination conditions, suggesting that the DEM representation preserves strong geometric cues. VGGT-SLAM++ supports both calibrated and uncalibrated versions. For completeness of the pipeline, we choose to report results with uncalibrated version, while calibrated versions are discussed in Appendix A1.

However, on uncalibrated grayscale (monochrome) datasets [5] (results shown inAppendix A1), VGGT-SLAM++ underperforms relative to classical SLAM pipelines whose front-end odometry relies on feature tracking or optical flow rather than RGB-based transformer inference of VGGT originally trained on RGB datasets. Even in these challenging cases, the proposed method consistently improves upon VGGT-SLAM for both the $\mathrm{Sim}(3)$ and $\mathrm{SL}(4)$ formulations. For several long paths of the KITTI Odometry dataset [20] monochrome sequences [5], the $\mathrm{SL}(4)$ variant of VGGT-SLAM failed to converge, further underscoring the robustness and stability of the our framework. Best is green, 2nd is light-green, 3rd is yellow.

Table 1. KITTI Odometry Benchmark. Absolute trajectory RMSE error (ATE, meters). Gray shade indicates results from classical methods. "–" indicates SL(4) does not converge.

Method	Uncalib.	00	01	02	03	04	05	06	07	08	09	10	Avg.
Classical feature-based SLAM													
ORB-SLAM2 (w/o LC) [38]	✗	40.65	502.20	47.82	**0.94**	1.30	29.95	40.82	16.04	43.09	38.77	**5.42**	69.73
ORB-SLAM2 (w/ LC) [38]	✗	**6.03**	508.34	**14.76**	1.02	1.57	**4.04**	**11.16**	**2.19**	**38.85**	**8.39**	6.63	54.82
LDSO [19]	✗	9.32	**11.68**	31.98	2.85	**1.22**	5.10	13.55	2.96	129.02	21.64	17.36	**22.43**
Learning-based SLAM													
DROID-SLAM [61]	✗	92.10	5344.60	107.61	2.38	1.00	118.50	62.47	21.78	161.60	72.32	118.70	554.82
DPV-SLAM [31]	✗	112.80	11.50	123.53	2.50	0.81	57.80	54.86	18.77	110.49	76.66	13.65	53.03
DPV-SLAM++ [31]	✗	8.30	11.86	39.64	2.50	0.78	5.74	11.60	1.52	110.90	76.70	13.70	25.75
VGGT-SLAM (Sim(3)) [36] [7]	✓	125.11	120.96	288.82	5.16	0.96	29.92	15.03	14.58	235.80	38.44	18.60	81.22
VGGT-SLAM (SL(4)) [36] [7]	✓	–	157.01	–	28.82	0.98	–	–	–	250.72	–	122.32	N/A
VGGT-SLAM++ (Ours) [7]	✓	119.00	109.64	223.21	4.50	0.95	25.21	13.65	12.17	155.00	35.26	15.71	64.94

Table 2. TUM RGB-D Benchmark. Absolute trajectory RMSE error (ATE, meters). Gray shade indicates results from calibrated methods.

Method	Uncalib.	360	desk	desk2	floor	plant	room	rpy	teddy	xyz	**Avg.**
ORB-SLAM3 [6]	✗	**0.017**	0.210	–	**0.034**	–	–	–	**0.009**	N/A	N/A
DeepV2D [59]	✗	0.243	0.166	0.379	1.653	0.203	0.246	0.105	0.316	0.064	0.375
DeepFactors [10]	✗	0.159	0.170	0.253	0.169	0.305	0.364	0.043	0.601	0.035	0.233
DPV-SLAM [31]	✗	0.112	0.018	0.029	0.057	0.021	0.330	0.030	0.084	0.010	0.076
DPV-SLAM++ [31]	✗	0.132	0.018	0.029	0.050	0.022	**0.096**	0.032	0.098	0.010	0.054
GO-SLAM [72]	✗	0.089	**0.016**	0.028	0.025	0.026	0.052	**0.019**	0.048	0.010	**0.035**
DROID-SLAM [61]	✗	0.111	0.018	0.042	**0.021**	**0.016**	0.049	0.026	0.048	0.012	0.038
MASt3R-SLAM [40]	✗	0.049	0.016	**0.024**	0.025	0.020	0.061	0.027	0.041	**0.009**	**0.030**
DROID-SLAM* [61]	✓	0.202	0.032	0.091	0.064	0.045	0.918	0.056	0.045	0.012	0.180
MASt3R-SLAM* [40]	✓	0.070	0.035	0.055	0.056	0.035	0.118	0.041	0.114	0.020	0.062
VGGT-SLAM (Sim3) [36]	✓	0.123	0.040	0.055	0.254	0.022	0.088	0.041	0.032	0.016	0.079
VGGT-SLAM (SL4) [61]	✓	0.071	0.025	0.040	0.141	0.023	0.102	0.030	0.034	0.014	0.053
VGGT-SLAM++ (Ours)	✓	0.042	0.025	0.027	0.077	0.042	0.027	0.026	0.029	0.016	**0.036**

Table 3. 7-SCENES Benchmark. Absolute trajectory RMSE error (ATE, meters). Gray shade indicates results from calibrated methods.

Method	Uncalib.	chess	fire	heads	office	pumpkin	kitchen	stairs	Avg.
NICER-SLAM3 [75]	✗	**0.033**	0.069	0.042	0.108	0.200	**0.039**	0.108	0.086
DROID-SLAM [61]	✗	0.036	0.027	0.025	**0.066**	0.127	0.040	0.026	0.050
MASt3R-SLAM [40]	✗	0.053	**0.025**	**0.015**	0.097	**0.088**	0.041	**0.011**	**0.047**
DROID-SLAM* [61]	✓	0.047	0.038	0.034	0.136	0.166	0.080	0.044	0.078
MASt3R-SLAM* [40]	✓	0.063	0.046	0.029	0.103	0.114	0.074	0.032	0.066
VGGT-SLAM (Sim3) [36]	✓	0.037	0.026	0.018	0.104	0.133	0.061	0.093	0.067
VGGT-SLAM (SL4) [36]	✓	0.036	0.028	0.018	0.103	0.133	0.058	0.093	0.067
VGGT-SLAM++ (Ours)	✓	0.034	0.023	0.017	0.104	0.127	0.085	0.060	**0.064**

Table 4. Virtual KITTI Benchmark. RMSE ATE (m). Methods shown per sequence, across weather variants.

Method	Uncalib.	Clone	Fog	Morning	Overcast	Rain	Sunset	Avg.
Sequence 01								
DROID-SLAM [61]	✗	1.03	1.87	0.99	1.01	0.78	1.15	1.14
CUT3R [27]	✗	43.30	63.19	50.60	38.73	51.55	43.79	48.53
VGGT-SLAM (Sim(3)) [36]	✓	1.44	3.20	0.82	0.78	1.56	3.02	1.80
VGGT-SLAM (SL(4)) [36]	✓	3.32	9.02	1.46	1.74	5.99	6.21	4.62
VGGT-SLAM++	✓	1.03	3.06	1.28	1.68	2.43	3.28	2.13
Sequence 02								
DROID-SLAM [61]	✗	0.10	0.04	0.05	0.05	0.04	0.11	0.07
CUT3R [27]	✗	23.77	9.95	28.42	24.64	7.96	25.97	20.12
VGGT-SLAM (Sim(3)) [36]	✓	0.10	0.15	0.14	0.31	0.21	0.63	0.26
VGGT-SLAM (SL(4)) [36]	✓	0.098	0.15	0.30	0.21	0.21	0.63	0.27
VGGT-SLAM++	✓	0.10	0.15	0.14	0.31	0.21	0.60	0.18
Sequence 06								
DROID-SLAM [61]	✗	0.06	0.02	0.03	0.05	TL	0.02	0.04
CUT3R [27]	✗	0.84	0.41	0.60	0.72	1.06	1.01	0.77
VGGT-SLAM (Sim(3)) [36]	✓	0.10	0.54	0.14	0.82	0.28	0.93	0.47
VGGT-SLAM (SL(4)) [36]	✓	0.10	0.53	0.14	0.83	0.28	0.93	0.47
VGGT-SLAM++	✓	0.10	0.53	0.13	0.82	0.28	0.93	0.47
Sequence 18								
DROID-SLAM [61]	✗	2.48	2.03	1.89	2.33	2.55	1.94	2.20
CUT3R [27]	✗	19.44	8.63	6.72	20.21	16.78	31.12	17.15
VGGT-SLAM (Sim(3)) [36]	✓	0.50	0.98	0.25	2.57	2.00	0.36	1.11
VGGT-SLAM (SL(4)) [36]	✓	0.51	0.98	0.25	2.57	2.00	0.36	1.11
VGGT-SLAM++	✓	0.50	0.98	0.25	2.55	1.99	0.37	1.11
Sequence 20								
DROID-SLAM [61]	✗	3.59	5.08	3.73	3.85	3.78	4.90	4.16
CUT3R [27]	✗	129.50	76.96	117.95	114.51	66.70	116.53	103.69
VGGT-SLAM (Sim(3)) [36]	✓	3.00	8.45	6.41	10.00	6.84	3.64	6.39
VGGT-SLAM (SL(4)) [36]	✓	3.87	9.50	8.21	10.00	6.64	3.65	6.98
VGGT-SLAM++	✓	3.00	8.45	6.11	10.00	5.84	3.64	6.17

Compared to the VGGT-SLAM baseline (Sim(3)+SL(4) averaged per dataset), VGGT-SLAM++ reduces ATE by 20% on KITTI, 45% on TUM, 5% on 7-Scenes, 14% on Virtual KITTI, 9% on EuRoC [5] (see Appendix A1). The combined VGGT-SLAM baseline (Sim(3)+SL(4), averaged per-dataset) results in ATE RMSE 17.13 m whereas that of VGGT-SLAM++ is 13.94 m, across the four datasets, hence we achive an overall improvement by **18.6%**.

4.2. Ablations

Table 5 compares different DEM rendering choices for VGGT-SLAM++ on KITTI odometry [20] sequences 00–10 and reports their respective ATE RMSE (m). The best hyperparameter in every sequence is shaded in green and the second best hyperparameter is shaded light green. It has been observed that all the ablation techniques work equally well for certain sequences, hence no color shading has been done to indicate the best choice.

Let the DEM height aggregation be

$$H(x,y) = \mathrm{red}_\tau\left(\{h_{x,y}^{(k)}\}\right), \qquad I(x,y) = \mathcal{N}(H)(1 - \alpha_{\text{edge}}\|\nabla\mathcal{N}(H)\|). \tag{5}$$

Table 5. DEM hyperparameter ablations on KITTI Odometry.

Method	00	01	02	03	04	05	06	07	08	09	10	Avg.
VGGT-SLAM++ (default)	119.00	109.64	223.21	4.50	0.95	25.21	13.65	12.17	155.00	35.26	15.71	64.936
Mean reducer (no softmax)	120.49	109.64	223.21	4.50	0.95	25.21	13.65	12.17	155.00	35.26	15.71	65.072
Softmax ($\tau = 0.10$)	119.00	109.64	223.21	4.50	0.95	25.21	13.65	12.17	155.00	35.26	15.71	64.936
Softmax ($\tau = 0.005$)	120.49	109.64	219.25	4.50	0.95	25.21	13.65	12.17	155.00	35.26	15.71	64.711
Half resolution	116.53	109.64	171.42	4.50	0.95	20.79	5.85	12.17	155.00	35.26	15.71	58.893
High resolution	132.54	109.64	223.21	4.50	0.95	23.32	13.65	12.17	155.00	35.26	15.71	65.995
No edge-enhancement	120.49	109.64	219.25	4.50	0.95	25.21	13.65	12.17	155.00	35.26	15.71	64.711
Slight edge-enhancement	119.00	109.64	223.21	4.50	0.95	25.21	13.65	12.17	155.00	35.26	15.71	64.936

where red_τ denotes the reducer (mean, max, or softmax temperature τ), $\mathcal{N}(\cdot)$ is percentile normalisation, and α_{edge} controls Sobel-based edge shading. Resolution is determined by the meters-per-pixel parameter $\mathrm{mpp} = S/N_{\text{px}}$. Half/high resolution correspond to $N_{\text{px}} \in \{45k, 180k\}$ (default $90k$), while no/slight edge enhancement use $\alpha_{\text{edge}} \in \{0, 0.5\}$. [Refer to Appendix A3 for more information]

4.3. Discussion

Our method shows consistent improvements in both ATE and runtime efficiency while maintaining low memory usage. VGGT-SLAM++ achieves near real-time operation, confirming the benefits of spatially corrective optimization and DEM-based compactness.

An observation is the limited ability of the VGGT odometry module to provide accurate motion estimates on monochrome (grayscale) datasets such as **EuRoC** [5] (Tables and discussion in Appendix A1), as the underlying transformer was trained exclusively on RGB data. However, the proposed back-end framework significantly improves the Absolute Trajectory Error (ATE) by **18.6%** (across all five datasets), relative to the VGGT-SLAM baseline.

5. Conclusion

We presented **VGGT-SLAM++**, a transformer-based visual SLAM system that couples VGGT-derived odometry with a DEM-based covisibility framework and a local bundle adjustment. By representing each submap as a compact DEM, the system preserves essential structural cues of a scene while enabling efficient retrieval through DINOv2 embeddings. We achieve SOTA accuracy on RGB datasets and delivers notable improvements on monochrome sequences where feed-forward transformer odometry is less reliable. The DEM representation also provides accuracy gains with minimal computational cost, making the system well-suited for real-time deployment on edge platforms. Future work will explore model compression and multi-modal sensing to further improve computational burden, and generalization.

Acknowledgement. We thank Aryan Singh for assistance with some of the experiments. This research was conducted in collaboration with Addverb Technologies and IHFC.

References

[1] Bakhriddin Akhmedov. Using the fundamentals of the theory of measurement errors in performing geodesic measurement and calculation works. In *E3S Web of Conferences*, page 03012. EDP Sciences, 2023. 6

[2] Luca Barsellotti, Lorenzo Bianchi, Nicola Messina, Fabio Carrara, Marcella Cornia, Lorenzo Baraldi, Fabrizio Falchi, and Rita Cucchiara. Talking to dino: Bridging self-supervised vision backbones with language for open-vocabulary segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22025–22035, 2025. 5

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 2

[4] James V Burke and Michael C Ferris. A gauss—newton method for convex composite optimization. *Mathematical Programming*, 71(2):179–194, 1995. 6

[5] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016. 6, 7, 8

[6] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6): 1874–1890, 2021. 2, 7

[7] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37 (6):1874–1890, 2021. 7

[8] Shoubin Chen, Baoding Zhou, Changhui Jiang, Weixing Xue, and Qingquan Li. A lidar/visual slam backend with loop closure detection and graph optimization. *Remote sensing*, 13(14):2720, 2021. 2

[9] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, and Cyrill Stachniss. Overlapnet: Loop closing for lidar-based slam. *arXiv preprint arXiv:2105.11344*, 2021. 3

[10] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2):721–728, 2020. 7

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2

[12] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[13] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *IEEE Transactions on Big Data*, 2025. 4, 5

[14] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *2025 International Conference on 3D Vision (3DV)*, pages 1–10. IEEE, 2025. 3

[15] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. In *2012 IEEE international conference on robotics and automation*, pages 1691–1696. IEEE, 2012. 2

[16] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5

[17] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 6

[18] Dorian Galvez-Lopez and Juan D Tardos. Real-time loop detection with bags of binary words. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 51–58. IEEE, 2011. 2

[19] Xiang Gao, Rui Wang, Nikolaus Demmel, and Daniel Cremers. Ldso: Direct sparse odometry with loop closure. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2198–2204. IEEE, 2018. 7

[20] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11): 1231–1237, 2013. 6, 7, 8

[21] Sudarshan S Harithas, Gurkirat Singh, Aneesh Chavan, Sarthak Sharma, Suraj Patni, Chetan Arora, and Madhava Krishna. Findernet: A data augmentation free canonicalization aided loop detection and closure technique for point clouds in 6-dof separation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8399–8408, 2024. 2, 3

[22] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 2

[23] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3r: Empowering unconstrained 3d reconstruction with camera and scene priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1071–1081, 2025. 3

[24] Satoshi Kagami, Koichi Nishiwaki, James J Kuffner, Kei Okada, Masayuki Inaba, and Hirochika Inoue. Vision-based 2.5 d terrain modeling for humanoid locomotion. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, pages 2141–2146. IEEE, 2003. 5

[25] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava

Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 9(2):1286–1293, 2023. 2, 5

[26] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025. 3

[27] Ramil Khafizov, Artem Komarichev, Ruslan Rakhimov, Peter Wonka, and Evgeny Burnaev. G-cut3r: Guided 3d reconstruction with camera and depth prior integration. *arXiv preprint arXiv:2508.11379*, 2025. 8

[28] Yang Li, Jianke Zhu, Steven CH Hoi, Wenjie Song, Zhefeng Wang, and Hantang Liu. Robust estimation of similarity transformation for visual object tracking. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8666–8673, 2019. 6

[29] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9087–9098, 2023. 2

[30] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 17627–17638, 2023. 2

[31] Lahav Lipson, Zachary Teed, and Jia Deng. Deep patch visual slam. In *European Conference on Computer Vision*, pages 424–440. Springer, 2024. 7

[32] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 6

[33] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[34] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *ieee transactions on robotics*, 32(1):1–19, 2015. 2

[35] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, pages 674–679, 1981. 2, 3

[36] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025. 2, 3, 7, 8

[37] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635. IEEE, 2017. 2

[38] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 2, 7

[39] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2, 4

[40] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025. 3, 7, 8

[41] Tae Hyeon Nam, Jae Hong Shim, and Young Im Cho. A 2.5 d map-based mobile robot localization via cooperation of aerial and ground robots. *Sensors*, 17(12):2730, 2017. 5

[42] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, pages I–I. Ieee, 2004. 2

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3, 5

[44] Zhentian Qian, Jie Fu, and Jing Xiao. Towards accurate loop closure detection in semantic slam with 3d semantic covisibility graphs. *IEEE Robotics and Automation Letters*, 7(2):2455–2462, 2022. 2, 4

[45] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 2

[46] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2

[47] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011. 2

[48] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 6

[49] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 2

[50] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 3

[51] Cyrill Stachniss, John J Leonard, and Sebastian Thrun. Simultaneous localization and mapping. In *Springer handbook of robotics*, pages 1153–1176. Springer, 2016. 1

[52] Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993. 5

[53] Hauke Strasdat, J Montiel, and Andrew J Davison. Scale drift-aware large scale monocular slam. *Robotics: science and Systems VI*, 2(3):7, 2010. 2

[54] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 6

[55] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. Openvslam: A versatile visual slam framework. In *Proceedings of the 27th ACM international conference on multimedia*, pages 2292–2295, 2019. 2

[56] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 2

[57] Niko Sünderhauf and Peter Protzel. Towards a robust back-end for pose graph slam. In *2012 IEEE international conference on robotics and automation*, pages 1254–1261. IEEE, 2012. 2

[58] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6243–6252, 2017. 2

[59] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 7

[60] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 2

[61] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 6, 7, 8

[62] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. 1

[63] Hemant Tyagi, Elıf Vural, and Pascal Frossard. Tangent space estimation for smooth embeddings of riemannian manifolds®. *Information and Inference: A Journal of the IMA*, 2(1):69–114, 2013. 6

[64] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 2

[65] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024. 3

[66] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 2, 3

[67] Peide Wang. Research on comparison of lidar and camera in autonomous driving. In *Journal of Physics: Conference Series*, page 012032. IOP Publishing, 2021. 1

[68] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025. 3

[69] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 2

[70] Peipei Xia, Li Zhang, and Fanzhang Li. Learning similarity with cosine similarity ensemble. *Information sciences*, 307:39–52, 2015. 5

[71] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 2

[72] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3727–3737, 2023. 7

[73] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 7244–7251. IEEE, 2018. 6

[74] Xiangmo Zhao, Pengpeng Sun, Zhigang Xu, Haigen Min, and Hongkai Yu. Fusion of 3d lidar and camera data for object detection in autonomous vehicle applications. *IEEE Sensors Journal*, 20(9):4901–4913, 2020. 1

[75] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. In *2024 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2024. 2, 8