

VGGT-SLAM++ // Supplementary Material

Avilasha Mandal¹ Rajesh Kumar² Sudarshan Sunil Harithas³ Chetan Arora¹
¹Indian Institute of Technology Delhi ²Addverb Technologies ³Brown University

Appendix

Appendix provides additional mathematical and algorithmic details that underpin the design of VGGT-SLAM++. We focus on six components: (A1) VGGT-SLAM++ Result Discussion on Established and Custom Datasets (A2) Depth Thresholding and Removal of Far-Field Floaters (A3) Global DEM Construction and Colour Mapping, (A4) FAISS-HNSW as the Covisibility Retrieval Backbone, (A5) Use of AnyLoc on DEM Images, (A6) Choice of Sim(3) for Back-end Optimisation

A1. VGGT-SLAM++ Result Discussion on Established and Custom Datasets

We have performed the experiments with VGGT-SLAM++ over 5 established datasets, KITTI odometry [12], Virtual KITTI [10], TUM RGB-D [32], 7-Scenes [28], EuRoC-MAV [4]. Our corrective backend leads to drift correction with loop detections and closures.

An observation is the limited ability of the VGGT odometry module to provide accurate motion estimates on monochrome (grayscale) datasets such as **EuRoC** (shown in table 1; best result is green, second best is light-green, third best is yellow.), as the underlying transformer was trained exclusively on RGB data, hence we observe under-performance of VGGT-SLAM++ compared to classical methods and DROID SLAM [33]. Yet VGGT-SLAM++ reduces ATE by 9% on EuRoC compared to the VGGT-SLAM baseline (Sim(3)+SL(4) averaged), due to our drift corrector backend module.

We also extend our experiments over custom datasets with various ground truth sources, such as the custom dataset (Fig 5(F) from the main paper) with 287.381m long path length recording by the GoPro HERO10 camera with GPS groundtruth (with precision 2m) from the Geo Tracker mobile application (ATE RMSE 7.17 ± 2 m) as shown in Fig. 1. We have also conducted several experiments with custom recordings by the OAK-1 camera (Fig 5(B) from the main paper) with Humanoid robot kinematics groundtruth (ATE RMSE 0.02m) over path length of 1.8m and another case, also with the OAK-1 based camera recording (Fig 5(C) from the main paper) with Cobot forward kinematics groundtruth (ATE RMSE

Table 1. EuRoC MAV Benchmark. Absolute trajectory RMSE error (ATE, meters). Gray shade indicates results from calibrated methods. “—” indicates SL(4) does not converge.

Method	Uncalib.	MH01	MH02	MH03	MH04	MH05	Avg.
ORB-SLAM [26]	x	0.071	0.067	0.071	0.082	0.060	0.070
DSM [41]	x	0.039	0.036	0.055	0.057	0.067	0.051
ORB-SLAM3 [26]	x	0.016	0.027	0.028	0.138	0.072	0.056
DeepFactors [6]	✓	1.587	1.479	3.139	5.331	4.002	3.108
DROID-SLAM [33]	✓	0.013	0.014	0.022	0.043	0.043	0.027
VGGT-SLAM (Sim(3)) [23]	✓	1.740	2.890	2.270	3.390	4.400	2.938
VGGT-SLAM (SL(4)) [23]	✓	3.780	3.960	3.710	—	—	N/A
VGGT-SLAM++ (Ours)	✓	1.600	2.700	1.900	2.980	4.150	2.666

0.01m) over path length of 1.8m (a planar scene, showing the ability of DEMs to make the trajectory estimation accurate even in planar domain). We also show that the (Digital Elevation Maps) DEMs [14] can handle the loop detection while re-localising a place, even from opposite ends with a completely different front view as they are inherently based upon the top view geometry which is constant while approaching the place from either sides. The center point of the 8-shaped loop ((Fig 1(A) from the main paper)) is reached from opposite ends leading to different front-views but since DEMs are agnostic to this fact, with their property of rendering the canonical height map of the place, we can detect loops in a viewpoint invariant style.

VGGT-SLAM++ supports both calibrated and uncalibrated versions. On KITTI, known intrinsics yields marginal gains: Seq. 05 improves from 25.21 m to 25.20 m; Seq. 03 remains at 4.50 m. Our novelty lies in engineering a backend, which cuts drift at high cadence, agnostic to the fact of whether calibration exists or not.

A2. Depth Thresholding and Removal of Far-Field Floaters

A recurrent failure mode in dense transformer reconstructions is the presence of *far-field floaters* [37, 39]: points reconstructed at extremely large depth due to textureless sky, horizon regions, or ambiguous background surfaces. These points do not correspond to observable geometry and, if left unfiltered, produce high-elevation spikes in the DEM that violate the planar assumption and might introduce unstable gradients for both DINOv2 [27] em-

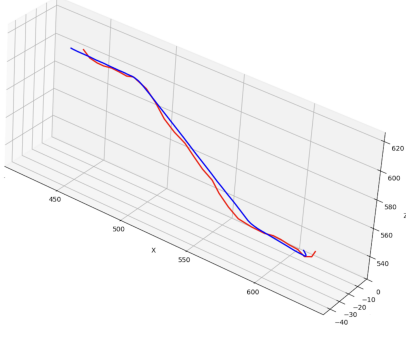


Figure 1. The red line is the ground truth reference from GPS readings and the blue line is the estimation by VGGT-SLAM++ for the custom GoPro camera dataset [Axes’ units are in meters].

beddings and the Sim(3) backend [31]. To prevent this, VGGT-SLAM++ applies a physically-motivated depth filter

$$\forall p_i = (x_i, y_i, z_i)^\top \in P$$

$$p_i \text{ “is kept” if } d_{\min} \leq \|p_i\|_2 \leq d_{\max}, \quad (1)$$

where d_{\min} and d_{\max} are user-specified bounds that remove implausibly near or implausibly distant structures. In practice, points with $\|p_i\|_2 \gg 30$ m typically originate from ambiguous sky pixels or regions with vanishing disparity; these inflate the DEM by acting as outlying “mountain peaks” during softmax aggregation [34]. Filtering them ensures that the retained set

$$P_{\text{valid}} = \{p_i \in P : d_{\min} \leq \|p_i\|_2 \leq d_{\max}\} \quad (2)$$

spans genuine scene geometry. This stabilises subsequent steps: (i) plane-fitting [9] becomes robust because extreme outliers no longer dominate the covariance; (ii) height aggregation behaves smoothly because all samples within a pixel correspond to metrically reasonable depths; and (iii) global DEM tiles exhibit clean, horizon-free elevation fields without sky-induced artefacts. Depth thresholding therefore plays the same role for height stability as confidence filtering does for prediction quality, ensuring that VGGT-SLAM++ builds DEMs solely from geometrically meaningful 3D structure.

A3. Global DEM Construction and Colour Mapping

This section explains construction of a global Digital Elevation Map (DEM) [14] from 3D points generated by a feedforward transformer [35]. The goal is to construct a planar-canonical DEM whose domain is a large rectangular region in a dominant ground-like plane and whose values encode signed height above that plane. The process consists of (i) fitting a stable reference plane, (ii) expressing all points in a canonical orthonormal frame,

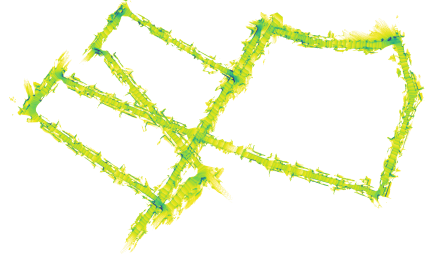


Figure 2. The DEM rendered from the 3D points aligned by odometry over the KITTI Sequence 05, with color mapping for better visualisation.

(iii) rasterising heights at a chosen spatial resolution, and (iv) feeding the grayscale DEM for DINOv2-based retrieval.

Input. Let

$$P = \{p_i\}_{i=1}^N, \quad p_i = (x_i, y_i, z_i)^\top \in \mathbb{R}^3, \quad (3)$$

denote all 3D points reconstructed by the frontend in a common world frame. These points arise from VGGT depth and Sim(3) odometry.

1. Plane fitting. Inherently a dominant structure underlies every practical scene from a standard dataset e.g. the ground plane, floor, or road is approximately planar over the global scale (comprising maximum number of points). We fit a plane

$$\Pi = \{p \in \mathbb{R}^3 : n^\top p + d = 0\}, \quad (4)$$

where $n \in \mathbb{R}^3$ is a unit normal and $d \in \mathbb{R}$ a signed offset. A RANSAC [9] loop proposes triples of points.

This yields a robust, metric ground plane even in cluttered scenes. It helps in robust loop detections even when robot re-visits a location from opposite ends (say approaching the same building (loop) from either it’s front or back side each of them with different views, would not be an issue while perceiving the bird’s eye view (BEV) [21] version of the place via height maps)

2. Canonical plane-aligned frame. We construct an orthonormal basis

$$R = [x \ y \ z] \in \text{SO}(3), \quad z := n, \quad (5)$$

where the in-plane axes x, y are determined by dominant eigenvectors of the projected points. Let origin $o = \bar{p}$ be the mean of all inlier points. Every world point is expressed in plane-aligned coordinates as

$$\tilde{p}_i = R^\top (p_i - o) = (u_i, v_i, h_i), \quad (6)$$

where (u_i, v_i) are planar coordinates and h_i the signed height above Π . All DEM operations use (u_i, v_i, h_i) .

3. Rasterisation into a metric grid. We seek a height field

$$H(u, v) : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}, \quad (7)$$

sampled on a regular grid with a *globally fixed* meters-per-pixel (mpp) resolution.

We first compute a robust planar bounding box of the projected points:

$$u_0 = \min_i u_i, \quad u_1 = \max_i u_i, \quad (8)$$

$$v_0 = \min_i v_i, \quad v_1 = \max_i v_i. \quad (9)$$

Let the longer span be

$$S = \max(u_1 - u_0, v_1 - v_0), \quad (10)$$

and choose a target number of pixels along this span, $\text{target_px_long} \in \mathbb{N}$. The global spatial resolution is then

$$\text{mpp} = \frac{S}{\text{target_px_long}}, \quad (11)$$

so each pixel, anywhere in the DEM, corresponds to exactly mpp meters in the plane.

Let W_{px} and H_{px} be the total grid width and height in pixels:

$$W_{\text{px}} = \left\lceil \frac{u_1 - u_0}{\text{mpp}} \right\rceil, \quad H_{\text{px}} = \left\lceil \frac{v_1 - v_0}{\text{mpp}} \right\rceil. \quad (12)$$

We tile this grid into $N_u \times N_v$ square tiles of fixed pixel size tile_px :

$$N_u = \left\lceil \frac{W_{\text{px}}}{\text{tile_px}} \right\rceil, \quad N_v = \left\lceil \frac{H_{\text{px}}}{\text{tile_px}} \right\rceil. \quad (13)$$

Thus each tile (I_u, I_v) covers a fixed metric region of size $\text{tile_px} \times \text{tile_px}$ pixels, i.e. $(\text{tile_px} \cdot \text{mpp}) \times (\text{tile_px} \cdot \text{mpp})$ square meters. The resolution mpp is global and does *not* change from tile to tile.

For each point (u_i, v_i) we first compute its global pixel coordinates

$$\hat{x}_i = \frac{u_i - u_0}{\text{mpp}}, \quad \hat{y}_i = \frac{v_i - v_0}{\text{mpp}}. \quad (14)$$

The corresponding tile indices and within-tile pixel indices are

$$I_u = \left\lfloor \frac{\hat{x}_i}{\text{tile_px}} \right\rfloor \quad (15)$$

$$I_v = \left\lfloor \frac{\hat{y}_i}{\text{tile_px}} \right\rfloor, \quad (16)$$

followed by clipping x, y into $[0, \text{tile_px} - 1]$. This is the logic implemented in the rasteriser: points are binned by tile (I_u, I_v) and then by integer pixel coordinates (x, y) inside each tile.

$$x = \text{round}(\hat{x}_i - I_u \cdot \text{tile_px}), \quad (17)$$

$$y = \text{round}(\hat{y}_i - I_v \cdot \text{tile_px}), \quad (18)$$

Height aggregation (the “reducer”). Multiple points may fall into the same pixel (x, y) of a tile. Let the set of heights for that pixel be

$$\{h_{x,y}^{(k)}\}_{k=1}^{K(x,y)}. \quad (19)$$

DEM construction applies a *reducer* function $\text{red}(\cdot)$ to obtain a single height value

$$H(x, y) = \text{red}\left(\{h_{x,y}^{(k)}\}_{k=1}^{K(x,y)}\right). \quad (20)$$

In practice we support three choices:

• **Mean reducer:**

$$\text{red}_{\text{mean}} = \frac{1}{K(x, y)} \sum_{k=1}^{K(x, y)} h_{x,y}^{(k)}. \quad (21)$$

This yields a smooth height field but can blur sharp steps.

• **Max reducer:**

$$\text{red}_{\text{max}} = \max_k h_{x,y}^{(k)}. \quad (22)$$

This preserves vertical discontinuities but is sensitive to outliers.

• **Softmax reducer** [34] (default):

$$\text{red}_{\text{softmax}} = \frac{\sum_k \exp(h_{x,y}^{(k)}/\tau) h_{x,y}^{(k)}}{\sum_k \exp(h_{x,y}^{(k)}/\tau)}, \quad (23)$$

where $\tau > 0$ is the *softmax temperature*. As $\tau \rightarrow 0$ the aggregation approaches the maximum (preserving sharp curbs and edges); as $\tau \rightarrow \infty$ it approaches the mean (smoother but more blurred). A small but non-zero τ provides a good compromise: sharp road geometry with reduced sensitivity to spurious height spikes.

Implementation-wise, the rasteriser builds per-pixel “buckets” of heights and applies the chosen reducer to each bucket. The raw DEM tile contains *the height field* (with NaNs (not a number) for empty pixels, independent of any visual colourisation). This height field is the signal used in the DINOv2-based retrieval pipeline.

4. Post-processing and Color-Map assigned to DEMs.

For consistent visual scaling across tiles we compute global DEM percentiles

$$h_{\min} = \text{perc}_{0.5}(H), \quad h_{\max} = \text{perc}_{99.5}(H), \quad (24)$$

and normalise

$$I_0(x, y) = \frac{\text{clip}(H(x, y), h_{\min}, h_{\max}) - h_{\min}}{h_{\max} - h_{\min}} \in [0, 1]. \quad (25)$$

NaN pixels (no observations) are displayed as pure white.

- **Edge enhancement.** Sobel gradients [29] ∇I_0 produce an edge mask

$$E = 1 - \alpha_{\text{edge}} \frac{\|\nabla I_0\|_2}{\text{perc}_{99}(\|\nabla I_0\|_2)}. \quad (26)$$

The grayscale image I_0 is passed to DINOv2 [27].

Here α_{edge} is the edge strength hyperparameter that determines how strongly high-gradient regions are darkened. Larger values produce heavier edge shading, while $\alpha_{\text{edge}} = 0$ disables the effect.

- **Hillshading** [18]. From the height map $H(x, y)$ we estimate local normals and compute standard Lambertian shading [5] with a virtual light direction ℓ :

$$S(x, y) = \max(0, n_{\text{surf}}(x, y)^\top \ell). \quad (27)$$

This reveals terrain-like structure. This colored version is only used for visualisations by humans and never used by DINOv2 unlike the grayscale version which it actually interpretes.

These operations produce the yellow-green (yellow is the ground plane, darker the shade of green higher the height of the real world point) DEM visualisations as shown in Fig. 2.

5. Discussion. Ablations with the hyper-parameters discussed in this section have been shown in Table ?? of main paper on KITTI odometry [12] sequences 00–10, reporting their respective average trajectory error in m (ATE RMSE). The default version of VGGT-SLAM++ uses softmax temperature $\tau = 0.02$ and edge strength hyperparameter $\alpha_{\text{edge}} = 0.95$, with 90k pixels DEM resolution and 4096 numbers of spatial tiles. the half resolution ablation study uses 45k pixels resolution and 2048 numbers of spatial tiles while the high resolution ablation study uses 180k pixels resolution and 4096 numbers of spatial tiles (the number of smaller tiles are same in all the three cases of the default, higher and lower resolutions). No edge enhancer implies $\alpha_{\text{edge}} = 0$ and the slight enhancement uses $\alpha_{\text{edge}} = 0.50$.

The results show that the reduction of the softmax temperature τ from 0.02 to 0.005 has lower overall ATE RMSE, as the edges are preserved, while keeping the smoothness intact at a lower τ . The half resolution scenario indicates presence of a trade-off in terms of the meters per pixel (mpp) represented in the DEM. The ATE RMSE decreases from a 90k pixels resolution (lower mpp) to a 45k pixels (higher mpp) at half resolution choice, indicating presence of a sweet spot of resolution during the height map rendering that leads to the best results as evident from the DEM ablation study.

A4. FAISS-HNSW as the Covisibility Retrieval Backbone

Modern SLAM backends increasingly rely on high-dimensional embeddings (e.g. DINOv2 [27] features) to establish covisible submaps or long-range loop closures. Nearest-neighbour search [19] in such spaces is the core operation: given database vectors $\{x_i\}_{i=1}^N \subset \mathbb{R}^d$ and a query $q \in \mathbb{R}^d$, one seeks

$$\arg \min_{i=1, \dots, N} \|q - x_i\|_2 \quad (28)$$

or equivalently top- k neighbours under L2 [3, 24] or cosine similarity [20, 40].

For moderate N this is feasible by brute force, but for typical SLAM settings (N raises to tens of thousands) [36] and queries arrive for every submap to be inserted; speed of exact search diminishes. Hence, approximate Nearest Neighbour Search (ANNS) [1] is required.

FAISS (Facebook AI Similarity Search) [8] is a widely used library that implements a large family of ANNS algorithms [1], unified under a common indexing abstraction. It does *not* learn features, but maintains a distributed service, or manage transactions; it provides efficient, well-engineered vector indices supporting (i) L2 distance, (ii) cosine similarity, (iii) inner product, (iv) CPU/GPU implementations, and (v) extremely fast incremental updates. This section summarises the mathematical foundations relevant to VGGT-SLAM++, before motivating our choice of the HNSW index [38].

Exact vs. approximate search. Exhaustive search computes all d -dimensional distances,

$$D_i = \|q - x_i\|_2^2, \quad i = 1, \dots, N, \quad (29)$$

which costs $\mathcal{O}(Nd)$ operations per query. This becomes prohibitive when N is large or queries arrive at video frequency. ANNS algorithms reduce this to *sublinear* complexity (typically $\mathcal{O}(\log N)$ or $\mathcal{O}(N^\rho)$ for $\rho < 1$) by replacing the full database with a compressed or navigable surrogate.

Approximation quality is measured by *recall*:

$$\text{recall}@k = \frac{|\text{ANN}(q, k) \cap \text{GT}(q, k)|}{k}, \quad (30)$$

where GT denotes exact top- k neighbours. For SLAM it is essential that recall [2] is high (strong covisibility cannot be missed), while latency must remain tightly bounded.

A5. Use of AnyLoc on DEM Images

AnyLoc [16] is a DINOv2-based [27] visual place recognition system that operates on standard images without task-specific retraining. DEMs [14] encode height which

constitute coherent images with stable local structure: edges, ridges, planar regions, and junctions appear as distinctive textures to the ViT backbone. Because DINOv2 features are largely appearance-agnostic and sensitive to both geometrical and semantic cues, the same descriptor (DINOv2) that matches natural images across viewpoints and illumination changes, also has the potential to match DEM tiles across traversal direction.

This compatibility allows us to use AnyLoc directly on the DEM domain: DEM for both tiles and query chips are passed through the same DINOv2 encoder. The resulting descriptors provide robust correspondences in both indoor and outdoor trajectories, serving as candidates for a visual place recognition using retrieval technique.

A6. Choice of Sim(3) for Back-end Optimisation

Let each VGGT-SLAM++ submap be represented by camera poses $\{\mathbf{T}_{w \leftarrow c}^{(i)}\}_i$ and dense point maps $\{P^{(i)}\}_i \subset \mathbb{R}^3$ as outputted by VGGT [35], all expressed in a common but *unknown* metric scale. In a purely projective formulation, one would relate two submaps via a 4×4 projective warp $\mathbf{H} \in \text{SL}(4)$ [15],

$$\mathbf{H} \in \text{SL}(4) = \{\mathbf{H} \in \mathbb{R}^{4 \times 4} : \det(\mathbf{H}) = 1\} \quad (31)$$

which can encode non-uniform scaling, shear, and general projective skew. In the classical setting, this is needed because monocular reconstructions are projectively ambiguous: points $x, x' \in \mathbb{P}^3$ satisfy $\tilde{x}' \sim \mathbf{H}\tilde{x}$ and \mathbf{H} is estimated from a homogeneous system $\mathbf{A}h = 0$, $h = \text{vec}(\mathbf{H})$, by taking the right singular vector of \mathbf{A} associated with the smallest singular value [30].

In VGGT-SLAM++, this level of freedom is both unnecessary and harmful. The frontend explicitly enforces parallax: keyframes are selected only when their disparity [17] exceeds a threshold, so successive submaps are linked by viewpoints with a non-trivial baseline. Hence the regime where pure projective ambiguity is severe (extremely small baselines [11], near-planar scenes) gets inherently avoided. So solving for the 7 degrees of freedom is sufficiently okay for an affine solution, as projective ambiguity hardly creeps in due to the chosen setting as discussed.

By restricting the backend to Sim(3) [13], we instead solve a well-posed, over-constrained problem on a 7-dimensional Lie group [22]. The parameter vector $\xi \in \mathbb{R}^7$ in Sim(3) directly encodes observable quantities, so the associated Jacobian [25] has a small, well-understood gauge nullspace and a spectrum whose dominant directions correspond to real geometric corrections. Intuitively, accumulated error over 7 meaningful degrees of freedom is far easier to stabilise than over 15 largely redundant ones in SL(4) [7, 15] optimisation problem.

SL(4) did not converge in long KITTI and EuRoC sequences evident in the tables 1 from main paper (KITTI) and 1 (EuRoC). Empirically, Sim(3) based back-end, yields substantially less drift and robust convergence on all KITTI and EuRoC trajectories, providing both physical and numerical justification for choosing Sim(3) over SL(4) in VGGT-SLAM++.

References

- [1] Jeffrey S Beis and David G Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pages 1000–1006. IEEE, 1997. 4
- [2] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994. 4
- [3] Peter Bühlmann and Bin Yu. Boosting with the ℓ_2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003. 4
- [4] 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. 1
- [5] Chi Kin Chow and Shiu Yin Yuen. Recovering shape by shading and stereo under lambertian shading model. *International journal of computer vision*, 85(1):58–100, 2009. 4
- [6] Jan Czarowski, 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. 1
- [7] Simon Kirwan Donaldson and Peter B Kronheimer. *The geometry of four-manifolds*. Oxford university press, 1997. 5
- [8] 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
- [9] 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. 2
- [10] 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. 1
- [11] David Gallup, Jan-Michael Frahm, Philippos Mordohai, and Marc Pollefeys. Variable baseline/resolution stereo. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 5
- [12] 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. 1, 4

- [13] W Nicholas Greene, Kyel Ok, Peter Lommel, and Nicholas Roy. Multi-level mapping: Real-time dense monocular slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 833–840. IEEE, 2016. 5
- [14] 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. 1, 2, 4
- [15] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5
- [16] 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. 4
- [17] Petr Kellnhofer, Tobias Ritschel, Karol Myszkowski, and Hans-Peter Seidel. Optimizing disparity for motion in depth. In *Computer Graphics Forum*, pages 143–152. Wiley Online Library, 2013. 5
- [18] Patrick J Kennelly and A Jon Kimerling. Hillshading of terrain using layer tints with aspect-variant luminosity. *Cartography and Geographic Information Science*, 31(2): 67–77, 2004. 4
- [19] Ashraf Masood Kibriya. *Fast algorithms for nearest neighbour search*. PhD thesis, The University of Waikato, 2007. 4
- [20] Alfina Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International conference on cyber and IT service management*, pages 1–6. IEEE, 2016. 4
- [21] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, et al. Delving into the devils of bird’s-eye-view perception: A review, evaluation and recipe. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2151–2170, 2023. 2
- [22] Alexander Vasil’evich Loboda. On 7-dimensional lie algebras admitting levi-nondegenerate orbits in $\mathfrak{c} 4$. *Trudy Moskovskogo Matematicheskogo Obshchestva*, 84(2):205–230, 2023. 5
- [23] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025. 1
- [24] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, and Zhen Wang. Multi-class generative adversarial networks with the l2 loss function. *arXiv preprint arXiv:1611.04076*, 5:1057–7149, 2016. 4
- [25] James S Milne. Jacobian varieties. In *Arithmetic geometry*, pages 167–212. Springer, 1986. 5
- [26] 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. 1
- [27] 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. 1, 4
- [28] 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. 1
- [29] Irwin Sobel, Gary Feldman, et al. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, 1968:271–272, 1968. 4
- [30] Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993. 5
- [31] 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
- [32] 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. 1
- [33] 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. 1
- [34] Haolun Wang, Tahir Mahmood, and Kifayat Ullah. Improved cocoso method based on frank softmax aggregation operators for t-spherical fuzzy multiple attribute group decision-making. *International Journal of Fuzzy Systems*, 25(3):1275–1310, 2023. 2, 3
- [35] 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, 5
- [36] Ke Wang, Juwei Guo, Kai Chen, and Jianbo Lu. An in-depth examination of slam methods: Challenges, advancements, and applications in complex scenes for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2025. 4
- [37] Tristan Wirth, Arne Rak, Volker Knauth, and Dieter W Fellner. A post processing technique to automatically remove floater artifacts in neural radiance fields. In *Computer Graphics Forum*, page e14977. Wiley Online Library, 2023. 1
- [38] Wentao Xiao, Yueyang Zhan, Rui Xi, Mengshu Hou, and Jianming Liao. Enhancing hnsf index for real-time updates: Addressing unreachable points and performance degradation. *arXiv preprint arXiv:2407.07871*, 2024. 4
- [39] WeiChen Yang, JinLong Shi, SuQin Bai, Qiang Qian, Zhen Ou, Dan Xu, Xin Shu, and YunHan Sun. Clear-pixels: Floaters free radiance fields without neural networks. *Knowledge-Based Systems*, 299:112096, 2024. 1

- [40] Jun Ye. Cosine similarity measures for intuitionistic fuzzy sets and their applications. *Mathematical and computer modelling*, 53(1-2):91–97, 2011. [4](#)
- [41] Jon Zubizarreta, Iker Aguinaga, and Jose Maria Martinez Montiel. Direct sparse mapping. *IEEE Transactions on Robotics*, 36(4):1363–1370, 2020. [1](#)