

MOGS: Monocular Object-guided Gaussian Splatting in Large Scenes

Shengkai Zhang¹, Yuhe Liu¹, Jianhua He², Xuedou Xiao¹, Mozi Chen^{1,*}, Kezhong Liu¹

Abstract—Recent advances in 3D Gaussian Splatting (3DGS) deliver striking photorealism, and extending it to large scenes opens new opportunities for semantic reasoning and prediction in applications such as autonomous driving. Today’s state-of-the-art systems for large scenes primarily originate from LiDAR-based pipelines that utilize long-range depth sensing. However, they require costly high-channel sensors whose dense point clouds strain memory and computation, limiting scalability, fleet deployment, and optimization speed. We present MOGS, a monocular 3DGS framework that replaces active LiDAR depth with object-anchored, metrized dense depth derived from sparse visual-inertial (VI) structure-from-motion (SfM) cues. Our key idea is to exploit image semantics to hypothesize per-object shape priors, anchor them with sparse but metrically reliable SfM points, and propagate the resulting metric constraints across each object to produce dense depth. To address two key challenges, *i.e.*, insufficient SfM coverage within objects and cross-object geometric inconsistency, MOGS introduces 1) a multi-scale shape consensus module that adaptively merges small segments into coarse objects best supported by SfM and fits them with parametric shape models, and 2) a cross-object depth refinement module that optimizes per-pixel depth under a combinatorial objective combining geometric consistency, prior anchoring, and edge-aware smoothness. Experiments on public datasets show that, with a low-cost VI sensor suite, MOGS reduces training time by up to 30.4% and memory consumption by 19.8%, while achieving high-quality rendering competitive with costly LiDAR-based approaches in large scenes. The source code is publicly available at <https://github.com/ClarenceZSK/MOGS/>.

I. INTRODUCTION

3D Gaussian Splatting (3DGS) has recently achieved real-time, photorealistic view synthesis with impressive fidelity [1]–[3]. When extended to large scenes, such high-fidelity mapping holds substantial potential for advanced environment sensing and reasoning, enhancing intelligent applications such as autonomous driving. Among existing approaches, LiDAR-initialized 3DGS is particularly effective at scale thanks to metrically accurate ranging and robustness across textures [4]–[7]. However, the high-channel LiDAR typically required is costly and produces dense point clouds; using these clouds to seed Gaussians balloons the number of primitives, inflating memory footprints and slowing training. At the city scale, these hardware and computational burdens constrain fleet deployment and hinder rapid iteration on algorithms and maps. Consequently, despite its accuracy, LiDAR-enabled 3DGS faces practical scalability limits in large scenes.

Authors¹ are with State Key Laboratory of Maritime Technology and Safety, Wuhan University of Technology, Wuhan, China. {shengkai, yuheliu, xuedouxiao, chenmz, kzliu}@whut.edu.cn

Author² is with University of Essex, Colchester, U.K. j.he@essex.ac.uk

*Corresponding author: Mozi Chen (chenmz@whut.edu.cn).

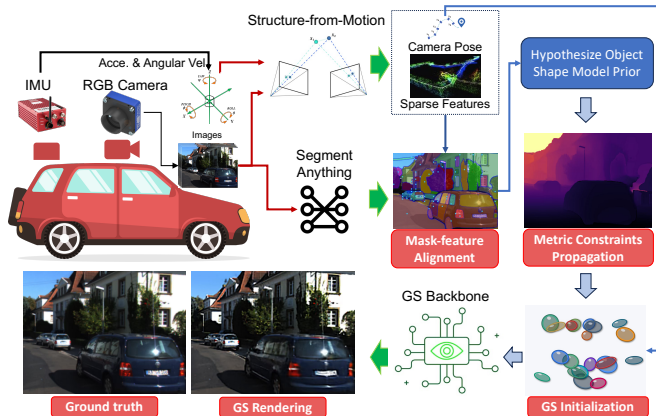


Fig. 1: MOGS exploits the rich semantics of RGB images and sparse, metrically reliable SfM cues to infer object-level shape priors. By propagating these metric constraints through shape priors, it produces metrized dense depth, approaching the effect of costly high-channel LiDAR, thereby enabling better Gaussian initialization and higher-quality rendering.

This motivates a monocular pathway to 3DGS in large scenes that is low-cost and unconstrained in range. The primary barrier to monocular 3DGS is the lack of reliable metric depth to initialize Gaussians, leading to scale drift and geometric inconsistency in large scenes [8]–[11]. We propose MOGS, a monocular 3DGS framework for large scenes with a visual-inertial (VI) structure-from-motion (SfM) frontend as shown in Fig. 1. At its core is an object-anchored metrization strategy that leverages image semantics to hypothesize shape priors, aligns them with sparse but metrically reliable SfM cues [12], [13], and propagates the resulting metric constraints across each object. Thus, we amplify sparse anchors into dense depth support.

However, realizing this idea poses two significant challenges. The first challenge is insufficient SfM coverage within objects. Monocular SfM yields tracks that cluster on corners and high-texture boundaries, leaving the interiors of large, low-texture surfaces, *e.g.*, roads, sky-occluded roofs, glass, sparsely or unevenly anchored, especially at long range. Occlusions and view-dependent appearance further reduce co-visibility, so per-object geometry becomes under-constrained by a handful of edge points, making any interior shape estimate ill-posed. The other challenge is enforcing geometric consistency across objects. Even when individual objects admit plausible local geometry, their relative placement, such as parallelism, coplanarity, contact with the ground plane, and mutual occlusion ordering, remains weakly constrained by monocular cues. Minor per-object scale errors and segmentation boundary jitter accumulate across views, resulting in inter-object depth offsets and broken discontinuities that

violate the global structure.

MOGS consists of two designs to address these challenges.

First, we propose a multi-scale shape consensus module. Starting from Segment Anything [14] masks, the module iteratively merges small segments that lack SfM support until each region contains sufficient features. Then it fits a compact set of parametric models, *e.g.*, plane, cylinder, and ellipsoid, and chooses the one with the maximal SfM agreement. The selected model propagates metric depth to all pixels in the region, yielding dense, object-consistent priors with confidence from fit quality. Regions without reliable parametric consensus, *e.g.*, foliage and cluttered surfaces, bypass modeling and are refined in the subsequent per-pixel optimization.

Second, we introduce a cross-object depth refinement module that refines per-object depth with a three-term combinatorial objective to enforce inter-object consistency. The objective comprises 1) a geometric consistency term that encourages agreement between the propagated metrized depth and the dense monocular depth from a large foundation model (LFM), Depth Anything [15], [16] up to a learned scale; 2) an LFM prior anchoring term that softly penalizes deviations between the refined depth and the LFM estimate, serving as a local-shape prior in weakly constrained or nonparametric regions without overriding metric alignment from other terms; and 3) an edge-aware smoothness term that preserves discontinuities while denoising interiors. Optimizing these terms per object implicitly couples neighboring regions through boundary constraints, yielding a metrically coherent depth field for 3DGS in large scenes.

Contributions. MOGS makes three contributions:

- We propose a multi-scale shape consensus module that establishes object-level shape models that agree with sparse SfM cues and converts them into dense, metrized depth priors for all pixels in each object, enabling reliable Gaussian initialization in large scenes.
- We develop a cross-object depth refinement module that optimizes per-object depth with a three-term combinatorial objective, *i.e.*, geometric consistency, LFM prior anchoring, and edge-aware smoothness, to align neighboring objects and produce a globally coherent depth field.
- Extensive experiments on public datasets demonstrate that MOGS reduces training time by up to 30.4% and memory consumption by 19.8%, while achieving rendering quality competitive with costly LiDAR-based approaches in large scenes.

II. RELATED WORK

The rise of 3D Gaussian Splatting (3DGS) has enabled real-time neural rendering by representing scenes as collections of oriented Gaussian ellipsoids that can be efficiently rasterized [1]. While early 3DGS methods achieved impressive fidelity on bounded scenes [17], [18], recent work has pushed toward large-scale reconstruction where scalability and robustness are critical [19], [20]. While these approaches commonly partition scenes to maintain real-time performance,

they typically require precomputed camera poses and rely on depth priors for stable initialization. In large scenes, obtaining such priors is challenging in that SfM can fail under wide baselines, sparse textures, or repetitive patterns [12], [21].

To address reliability at scale, a growing body of work fuses LiDAR with cameras to inject robust geometry and stabilize poses [4]–[6], [22]. LIV-GaussMap [4] initializes surface Gaussians and frame poses from a LiDAR-inertial backbone, then refines the map with image photometric gradients for real-time rendering across large environments. TCLC-GS [5] constructs a hybrid explicit and implicit representation, supervising splats with dense mesh depth to achieve fast training and high FPS rendering on Waymo and nuScenes. Gaussian-LIC [6] integrates LiDAR-IMU-camera SLAM with Gaussian mapping, initializing Gaussians from both LiDAR and triangulated visual points to reach real-time, photorealistic SLAM in unbounded scenes. Collectively, these systems demonstrate that LiDAR-guided 3DGS scales well and renders efficiently in large scenes. However, dense depth supervision over long ranges often demands high-channel LiDAR, which is costly (thousands of USD), and the resulting dense point clouds inflate memory footprints and slow GS optimization, limiting scalability and iteration speed in practice.

To avoid the cost and computational burden of high-channel LiDAR, recent works have explored monocular pathways for large-scale 3DGS. A typical approach is to leverage LFM for monocular depth estimation, such as Depth Anything [15], [16], to produce dense depth maps that initialize Gaussians. ICP-3DGS [23] couples LFM monocular depth with ICP to estimate poses and expand scenes for unbounded environments. Mode-GS [24] parameterizes anchors to calibrate scales of LFM monocular depth to bootstrap rendering. Complementary efforts in mono-GS SLAM [8], [11], [25] use monocular metric depth [26] during tracking and mapping to keep global structure consistent over long trajectories. DepthSplat [27] underscores the bidirectional benefit of enforcing photometric and geometric consistency when scaling to large scenes. Nonetheless, these methods rely on fine-tuned LFM tailored to specific datasets, which limits generalization.

Our object-guided approach fuses the strengths of geometrically consistent, up-to-scale LFM depth with metrically anchored sparse SfM cues. We exploit image semantics to infer object-level shape models and propagate sparse depth across each object, computing dense guidance without storing dense point clouds, thereby reducing memory consumption and computational complexity. Additionally, we utilize the geometric consistency of LFM depth during refinement to enhance accuracy and cross-object coherence in the reconstructed scene.

III. SYSTEM DESIGN OF MOGS

A. Overview

MOGS enables low-cost and efficient high-definition (HD) 3DGS rendering in large scenes with a VI-SfM frontend, *e.g.*, VINS [12]. As shown in Fig. 2, for each input image we first obtain semantic object masks using Segment Anything [14], then associate the 3D positions of SuperPoint features [28]

from the frontend with the corresponding masks. The multi-scale shape consensus module uses these associations to establish object-level shape models and propagate SfM cues, producing dense metrized depth for all pixels. This depth is initially coarse due to errors remain in nonparametric regions that do not admit standard models and near object boundaries. We therefore introduce a cross-object depth refinement module that further optimizes the depth with a three-term combinatorial objective. It consists of 1) a geometric consistency term that aligns the propagated metrized depth with scale-ambiguous but geometrically consistent dense LFM depth [16]; 2) an LFM prior anchoring term that softly penalizes deviations between the refined depth and the LFM estimate, serving as a local-shape prior in weakly constrained or nonparametric regions without overriding metric alignment from other terms; and 3) an edge-aware smoothness term that preserves discontinuities while denoising interiors. The refined dense depth then initializes Gaussians and bootstraps 3DGS optimization and rendering.

B. Multi-scale Shape Consensus

Sparse SfM features [12] rarely cover the interiors of large, low-texture objects *e.g.*, glass, roads, leaving their geometry under-constrained and depth noisy at long range. Pixel-wise refinement alone cannot reliably inject metricity into these regions, while naive semantic segments are often over-segmented and too small to accumulate sufficient SfM support. This module reframes the problem at the object level, where simple parametric priors, *e.g.*, plane, cylinder, and ellipsoid, can carry metric constraints across many pixels once a minimal set of reliable SfM anchors exists.

A multi-scale merging strategy is therefore essential. It adaptively fuses adjacent fine segments until each candidate object amasses enough SfM evidence to hypothesize and validate a shape, avoiding both over- and under-segmentation. Establishing such object-level shapes enables robust propagation of metric depth to all pixels within the object, producing dense metrized depth that stabilizes Gaussian initialization, suppresses floaters, and improves long-range fidelity.

We denote the 3D positions of SuperPoint features [28] from our SfM frontend [12] as $\{\mathbf{P}_i \in \mathbb{R}^3\}_{i=0,1,\dots,N}$, with $\mathbf{P}_i = [x_i, y_i, z_i]$, where N is the total number of features and z_i is the depth of feature i in the current camera frame. Meanwhile, for each image, we obtain K fine-grained semantic object masks, $\{M_k\}_{k=1,\dots,K}$ using Segment Anything [14]. Let \mathbf{u}_i be the 2D projection of \mathbf{P}_i in this image. We assign each feature to the object whose mask contains its projection, *i.e.*, $l_i = k$ if $\mathbf{u}_i \in M_k$, and collect the associated SfM features per object as $\mathcal{F}_k = \{i \mid \mathbf{u}_i \in M_k\}$. Features not covered by any mask are treated as background.

Multi-scale object merging. For small or low-texture objects, SfM features often concentrate along discontinuous boundaries, making their shapes hard to recover. We therefore impose a minimum-support threshold $n_{min} = \max\{n_{plane}, n_{ellipsoid}, n_{cylinder}\}$, where n_{plane} , $n_{ellipsoid}$, and $n_{cylinder}$ are the minimum numbers of points needed to uniquely determine a plane, an ellipsoid, and a cylinder,

which are 3, 9, and 7, respectively. We deem a mask M_k under-supported if $|\mathcal{F}_k| < n_{min}$.

Under-supported masks seek to merge into adjacent regions to form a coarser object. Let \mathcal{N}_u be the set of masks adjacent to an under-supported mask M_u . We seek a neighbor $M_i \in \mathcal{N}_u$ that is depth-contiguous with M_u . For each mask M_i , we compute the median SfM depth \bar{z}_i over its features. A neighbor M_i is eligible if the depth gap $\Delta z_{i,u} = |\bar{z}_i - \bar{z}_u| \leq \tau_i$, $\tau_i = 0.1 \times \bar{z}_i$, *i.e.*, a 10% relative tolerance. This criterion prevents merging adjacent objects at clearly different elevations, *e.g.*, an object on a table versus the floor. Eligible neighbors are then considered as a candidate set \mathcal{C}_u for merging M_u .

Next, among the candidates \mathcal{C}_u , we select the mask M_c with the minimum depth gap $\Delta z_{c,u}$ and merge it with M_u to form a new region $M_{u'} = M_c \cup M_u$. Then we update the SfM feature set $\mathcal{F}_{u'} = \mathcal{F}_u \cup \mathcal{F}_c$ and test whether $|\mathcal{F}_{u'}| \geq n_{min}$. If not, recompute candidates for $M_{u'}$ and continue merging with its remaining adjacent regions.

Object shape consensus. After merging, each object has sufficient SfM features. For object k , we then fit the shape that explains the majority of points in \mathcal{F}_k . Algorithm 1 is the pseudocode of our object shape consensus.

We consider three canonical primitives, *i.e.*, plane, ellipsoid, and cylinder, parameterized as:

- Plane: $\mathbf{S}_1 = (\mathbf{n}, d)$ with $\mathbf{n} \in \mathbb{R}^3$ a unit normal ($\|\mathbf{n}\| = 1$) and $d \in \mathbb{R}$ the signed offset;
- Ellipsoid: $\mathbf{S}_2 = (\mathbf{c}, \mathbf{A})$ with $\mathbf{c} \in \mathbb{R}^3$ the center and $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ a symmetric positive-definite shape matrix;
- Cylinder: $\mathbf{S}_3 = (\mathbf{p}, \mathbf{u}, r)$ with $\mathbf{p} \in \mathbb{R}^3$ a point on the axis, $\mathbf{u} \in \mathbb{R}^3$ a unit axis direction ($\|\mathbf{u}\| = 1$), and $r > 0$ the radius.

We denote the set of canonical primitives as $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3\}$. For each $\mathbf{S} \in \mathcal{S}$, we fit the object's SfM points \mathcal{F}_k using minimal-sample RANSAC [29], which returns an inlier set and a confidence score $C_k(\mathbf{S})$. The final model is chosen by

$$S_k^* = \arg \max_{\mathbf{S} \in \mathcal{S}} C_k(\mathbf{S}). \quad (1)$$

Specifically, for each $\mathbf{S} \in \mathcal{S}$, in the RANSAC, we repeat sampling up to $maxIter$ and keep the hypothesis that maximizes the number of inliers for the object. We define the inlier by computing the distance of a point $\mathbf{P} \in \mathcal{F}_k$ to a model \mathbf{S} as

$$r_k(\mathbf{P} \mid \mathbf{S}) = \left| \frac{f_k(\mathbf{P}, \mathbf{S})}{\|\nabla f_k(\mathbf{P}, \mathbf{S})\|^2} \nabla f_k(\mathbf{P}, \mathbf{S}) \right|, \quad (2)$$

where $f_k(\mathbf{P}, \mathbf{S})$ is the implicit function of the shape model of object k parameterized by \mathbf{S} . Points \mathbf{P}_0 on the surface satisfy $f_k(\mathbf{P}_0, \mathbf{S}) = 0$. $\nabla f_k(\mathbf{P}, \mathbf{S})$ denotes its spatial gradient. Given a geometric error threshold T_s , the inlier set of object k is

$$\mathcal{I}_k(\mathbf{S}) = \{\mathbf{P} \in \mathcal{F}_k \mid r_k(\mathbf{P} \mid \mathbf{S}) < T_s\}. \quad (3)$$

Then the inlier ratio can be defined as $\gamma_k(\mathbf{S}) = \frac{|\mathcal{I}_k(\mathbf{S})|}{|\mathcal{F}_k|}$.

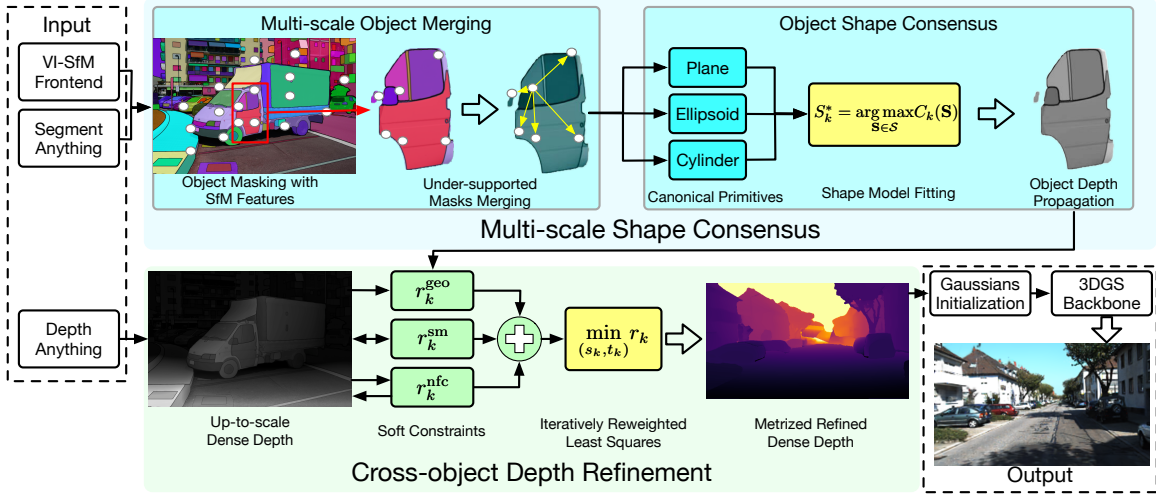


Fig. 2: System overview of MOGS. We align VI-SfM visual features with semantic masks from Segment Anything. The multi-scale shape consensus module then establishes object-level shape models and propagates SfM cues to produce metrized dense depth. Building on this, our cross-object depth refinement leverages LFM dense depth to further optimize pixel-wise estimates, yielding strong 3DGS initialization and, ultimately, higher-fidelity Gaussian splatting.

To compute the confidence score $C_k(\mathbf{S})$, we still need to know the residual variance over inliers as

$$\sigma_k^2(\mathbf{S}) = \frac{1}{|\mathcal{I}_k(\mathbf{S})|} \sum_{\mathbf{P} \in \mathcal{I}_k(\mathbf{S})} r_k(\mathbf{P} | \mathbf{S})^2. \quad (4)$$

We then define the confidence score, which rewards a high inlier ratio and a low residual variance, as

$$C_k(\mathbf{S}) = \gamma_k(\mathbf{S}) \exp\left(-\frac{\sigma_k^2(\mathbf{S})}{\beta T_s^2}\right), \quad (5)$$

where $\beta \in (0, 1]$ is a scale hyperparameter.

At this stage, depths for pixels on structured objects can be readily propagated from their fitted shape models. In contrast, points not explained by any canonical primitive typically lie in texture-rich regions, *e.g.*, foliage or cluttered surfaces. These pixels are sufficiently dense to seed Gaussians, and their depths are further refined in the subsequent module. In practice, the three canonical models cover most pixels in large scenes: on street-scene data, roughly 44% of segments (covering 73% of pixels) are assigned to our geometric structures.

C. Cross-object Depth Refinement

The depth propagated in the previous module can inherit SfM errors and shape-model bias, and it lacks cross-object constraints to preserve global structure. To address this, we incorporate the LFM, Depth Anything [16], which provides scale-ambiguous yet geometrically consistent dense depth from monocular images. We use this LFM depth as cross-object geometric supervision to refine the propagated depths.

Given a mask M_k with pixel domain Ω_k , we map the LFM's relative depth $\hat{d}(p)$ at pixel $p = (u, v)$ to absolute depth by an affine calibration:

$$\tilde{D}_k(p) = s_k \hat{d}(p) + t_k, \quad p \in \Omega_k, \quad (6)$$

where (s_k, t_k) are mask-wise scale and shift.

Let $D_k^{\text{model}}(p)$ be the depth at pixel p obtained by propagating the fitted canonical model for mask M_k , and let

Algorithm 1 Object Shape Consensus

```

1: Input: the SfM points  $\mathcal{F}_k$  on object  $k$ ,  $k = 1, 2, \dots, K$ .
2:  $S \leftarrow \emptyset$ 
3: for  $k = 1$  to  $K$  do
4:    $S_k^* \leftarrow \emptyset$ ,  $\mathcal{I}_k^{\text{best}} \leftarrow \emptyset$ ,  $C_k^{\text{best}} \leftarrow -\infty$ ,
5:   for each  $\mathbf{S} \in \mathcal{S}$  do
6:      $R_c \leftarrow \text{MINIMALSAMPLESIZE}(\mathbf{S})$ 
7:      $C_k(\mathbf{S}) \leftarrow -\infty$ ,  $\mathcal{I}_k^{\text{best}}(\mathbf{S}) \leftarrow \emptyset$ 
8:     for  $j = 1$  to  $\text{maxIter}$  do
9:        $r_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{F}_k, R_c)$ 
10:       $S_k \leftarrow \text{FITSTRUCTURE}(\mathbf{S}, r_k)$ 
11:       $\mathcal{I}_k(\mathbf{S}) \leftarrow \{\mathbf{P} \in \mathcal{F}_k \mid r_k(\mathbf{P} | \mathbf{S}) < T_s\}$ 
12:       $\text{score} \leftarrow \text{Eqn. (5)}$ 
13:      if  $\text{score} > C_k(\mathbf{S})$  then
14:         $C_k(\mathbf{S}) \leftarrow \text{score}$ ,  $S_k^* \leftarrow S_k$ 
15:         $\mathcal{I}_k^{\text{best}}(\mathbf{S}) \leftarrow \mathcal{I}_k(\mathbf{S})$ 
16:      end if
17:    end for
18:    if  $C_k(\mathbf{S}) > C_k^{\text{best}}$  then
19:       $S_k^* \leftarrow S_k$ ,  $\mathcal{I}_k^{\text{best}} \leftarrow \mathcal{I}_k^{\text{best}}(\mathbf{S})$ ,  $C_k^{\text{best}} \leftarrow C_k(\mathbf{S})$ 
20:    end if
21:  end for
22:  if  $|\mathcal{I}_k^{\text{best}}| \geq \tau \cdot |\mathcal{F}_k|$  then
23:     $S \leftarrow S \cup \{S_k^*, C_k^{\text{best}}\}$ 
24:  end if
25: end for
26: return  $S$ 

```

$C_k \in [0, 1]$ denote that model's confidence. We estimate the affine calibration (s_k, t_k) with three complementary terms evaluated over the same domain Ω_k . A robust Huber loss $\rho(\cdot)$ downweights outliers.

Geometric consistency residual. This term enforces agreement between the refined \tilde{D}_k and the model-propagated depth

D_k^{model} .

$$r_k^{\text{geo}} = \sum_{p \in \Omega_k} C_k \times \rho(\tilde{D}_k(p) - D_k^{\text{model}}(p)). \quad (7)$$

LFM prior anchoring residual. This term softly anchors $\tilde{D}_k(p)$ to the LFM estimate $\hat{d}(p)$, helping preserve local structure in regions with weak geometric support:

$$r_k^{\text{nfc}} = \sum_{p \in \Omega_k} \rho(\tilde{D}_k(p) - \hat{d}(p)). \quad (8)$$

Edge-aware smoothness residual. This term encourages intra-mask smoothness while respecting image gradients to preserve sharp object boundaries.

$$r_k^{\text{sm}} = \sum_{(p,q) \in \mathcal{N}_k} w_{pq} (\tilde{D}_k(p) - \tilde{D}_k(q))^2. \quad (9)$$

where \mathcal{N}_k is a pixel neighborhood over Ω_k and w_{pq} is edge-gated pairwise weight that depends on local image gradients.

$$w_{pq} = \exp\left(-\frac{\|\nabla I(p)\|_2^2 + \|\nabla I(q)\|_2^2}{2\sigma_I}\right), (p, q) \in \mathcal{N}_k \quad (10)$$

where $\nabla I(\cdot)$ denotes the image spatial gradient, and σ_I controls the aggressiveness of edge downweighting. $(p, q) \in \mathcal{N}_k$ restricts smoothing within mask M_k , preventing cross-mask diffusion. The weight $w_{pq} \in (0, 1]$ decays near strong gradients, yielding anisotropic smoothing.

For mask M_k , we optimize the following combinatorial objective:

$$\min_{(s_k, t_k)} r_k = \lambda_{\text{geo}} r_k^{\text{geo}} + \lambda_{\text{nfc}} r_k^{\text{nfc}} + \lambda_{\text{sm}} r_k^{\text{sm}}, \quad (11)$$

where the positive weights λ_{geo} , λ_{nfc} , and $\lambda_{\text{sm}} > 0$ balance the geometric consistency, LFM prior anchoring, and edge-aware smoothness terms, respectively. We set them empirically per scene. In outdoor driving experiments, we use $\lambda_{\text{geo}} = 1.0$, $\lambda_{\text{nfc}} = 0.7$, and $\lambda_{\text{sm}} = 0.1$. For scenes dominated by large planes and regular structures, λ_{geo} can be increased, *e.g.*, to 1.2. For low-texture, repetitive-texture, or cluttered scenes, we recommend raising λ_{nfc} , *e.g.*, to 0.9. For foggy or low-light conditions with noisy depth, increasing λ_{sm} , *e.g.*, to 0.2, helps suppress artifacts.

We solve (11) by iteratively reweighted least squares (IRLS) and compute the refined depth via (6).

IV. EXPERIMENTAL EVALUATION

A. Implementation and Experimental Setup

MOGS aims to produce efficient, reliable dense depth maps that enable stronger Gaussian initialization and high-fidelity 3DGS rendering in large scenes. Accordingly, we evaluate MOGS on depth accuracy and final rendering quality.

Implementation. All experiments run on a single NVIDIA GeForce RTX 4090 GPU and an Intel Xeon Platinum 8347C CPU (2.10 GHz). We detect SuperPoint features and match them with LightGlue; correspondences are filtered by USAC-PROSAC. Accepted keypoints carry persistent global track IDs for downstream reconstruction. Keyframes are triggered jointly by inlier count and parallax thresholds. We initialize

3D points via SVD triangulation and refine them with nonlinear least squares using analytic Jacobians.

Depth-aware instance masks are sorted far-to-near by median depth, refined with a 2-pixel neighborhood voting radius, small-component merging to adjacent structures. Given an LFM depth map r , each mask M_k is calibrated by an affine mapping $\tilde{D}_k = s_k r + t_k$. Masks with $\text{mean}(r) > \tau_{\text{sky}} = 100$ are treated as sky and clamped to a maximum depth. When sufficient SfM points are present, $\{s_k, t_k\}$ are estimated via IRLS. Note that dynamic features in a scene are identified and compensated by [30].

Datasets and Metrics. We use KITTI-Depth [31] collection that covers urban, residential, road, and campus traffic scenes. The raw sensor platform comprises two global-shutter stereo rigs, a Velodyne HDL-64E LiDAR, and an OXTS RT3003 high-precision IMU; cameras and LiDAR are synchronized at 10 Hz, and the IMU at 100 Hz. KITTI-Depth provides approximately 93 k semi-dense depth maps temporally aligned with RGB camera and LiDAR. In addition, we also use KITTI-360 [32] that contains large-scale suburban cruising (total distance 73.7 km, over 83 k frames). The platform carries a forward-looking perspective stereo pair plus fisheye cameras, a Velodyne LiDAR, and SICK push-broom 2D lasers, providing 360° perception with consistent semantic labels. Ground-truth camera poses are retrieved from the datasets or captured via a motion capture system like NOKOV using customized experimental platforms. Correspondingly, all ground-truth depth maps were temporally and spatially aligned with the captured images.

Depth accuracy is reported using Absolute Relative Error (*AbsRel*), Root Mean Squared Error (*RMSE*), and δ_1 (percentage of $\max(d^*/d, d/d^*) < 1.25$). For 3DGS, rendering quality is measured by Peak Signal-to-Noise Ratio (*PSNR*), Structural Similarity Index (*SSIM*), and Learned Perceptual Image Patch Similarity (*LPIPS*). Training efficiency is measured by optimization iterations and wall-clock time to reach a target quality level under a fixed budget, and computational cost is further characterized by the number of active Gaussian primitives and peak GPU memory footprint.

B. Performance Evaluation

Depth Accuracy. We compare against SOTA monocular-based depth estimation baselines *Depth Anything V2* [33], *Depth Pro* [34], *ZoeDepth* [35], and *Metric3D v2* [36]. All these methods infer dense depth from monocular images using a pre-trained LFM. In the evaluation, we use the ground truth in the dataset as a reference to compute the average metrics. Quantitative results are summarized in Table I, showing that MOGS attains consistently lower *AbsRel* and *RMSE*, and higher δ_1 . Notably, the metrized dense depth obtained by MOGS surpasses all LFM estimates without any pre-training.

Rendering Quality. We first measure how the depth-enabled initializations affect 3DGS convergence speed, memory consumption, and rendering quality. We then compare our method against state-of-the-art monocular 3DGS baselines for qualitative rendering in outdoor large scenes.

TABLE I: Monocular depth estimation on KITTI-Depth and KITTI-360.

Method	KITTI-Depth			KITTI-360		
	AbsRel↓	RMSE↓	δ_1 ↑	AbsRel↓	RMSE↓	δ_1 ↑
Depth Anything V2	0.070	2.62	0.958	0.080	2.99	0.948
Depth Pro	0.060	2.34	0.970	0.071	2.70	0.959
ZoeDepth	0.082	3.10	0.941	0.093	3.55	0.929
Metric3D v2	0.059	2.32	0.971	0.070	2.67	0.960
MOGS (Ours)	0.058	2.29	0.973	0.070	2.63	0.961

TABLE II: NVS quality and convergence under fixed 3DGS budgets.

Init	Iter.↓	Gauss. ($\times 10^6$)	PSNR↑	SSIM↑	LPIPS↓
Init-Rand	46k	1.67	19.7	0.936	0.332
Init-LFM-M	39k	1.53	21.1	0.951	0.288
Init-MOGS	32k	1.34	21.9	0.958	0.236

1) *GS initialization.* We adopt an improved version of MonoGS [9] as GS backend. We compare different depth-enabled initializations to bootstrap GS optimization and rendering, including *Init-Rand* that initializes with random depth, *Init-LFM-M* that provides a scale factor computed by aligning LFM to ground truth and then metrize the LFM depth, and *Init-MOGS*. We conduct this experiment in two rounds. First, we fix a target *PSNR* and run these methods to see the number of iterations and generated Gaussians (column 2-3 in Table II). *Init-MOGS* reduces up to 30.4% of iterations and 19.8% of Gaussians to achieve high-quality GS rendering. Second, we fix a target number of iterations and run to see the rendering quality (column 4-6 in Table II). *Init-MOGS* shows the best rendering quality in terms of *PSNR*, *SSIM*, and *LPIPS*.

In addition to the average metrics in Table II, we plot the Gaussians-iteration curve, illustrating how each method grows its Gaussians during training as shown in Fig. 3. *MOGS* exhibits the lowest slope, indicating the slowest growing speed.

Fig. 4 shows the number of iterations to the target *PSNR* (TT-*PSNR*) with respect to different datasets. We fixed a target view for each experiment, and we start counting iterations from its first visibility of the view until its rendering *PSNR* exceeds 20 dB. Across all sequences, *MOGS* attains the target rendering quality with the fewest iterations and the fewest active Gaussians (refer to Fig. 3), indicating faster anchoring of geometry at the correct scale, reduced over-parameterization and drift, and a better quality-efficiency trade-off. We attribute these gains to object-level metric constraints, which make the geometry converge quickly and shift optimization to appearance earlier.

2) *GS rendering.*

Unless stated otherwise, all runs share identical hyperparameters (training steps, learning rate, cap on visible Gaussians), data splits (held-out novel views), and image resolution. We compare *MOGS* with SOTA monocular-based GS approaches MonoGS [9] and DepthSplat [27], as well as a SOTA LiDAR-based GS approach, GS-LIVM [22], on our splits using their

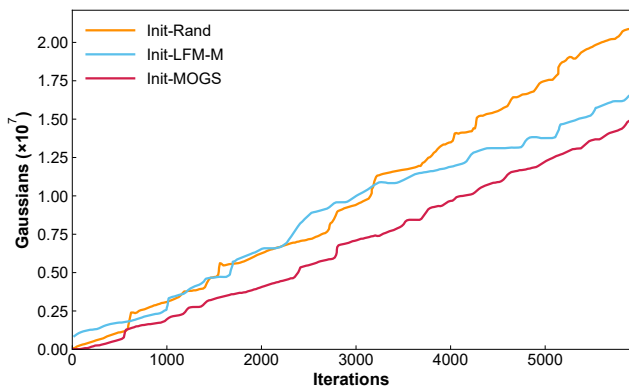
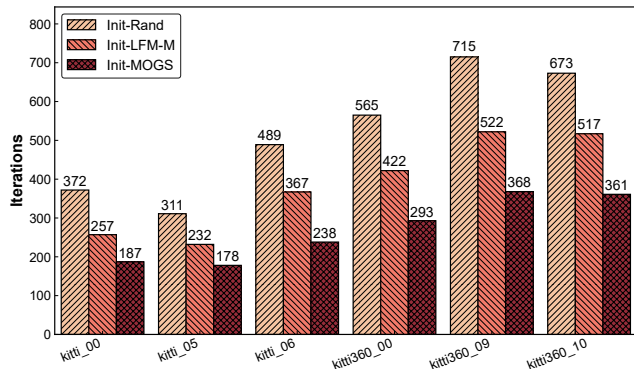


Fig. 3: Convergence and efficiency under different GS initializations, revealing how each method grows its Gaussians along with iterations.

Fig. 4: The numbers of iterations under different GS initializations that optimize a view rendering until 20 dB *PSNR* on multiple datasets.

recommended settings. When the depth is required, we use the method’s native inputs. We report rendering metrics under the same computational resource budget. Table III summarizes quantitative novel-view metrics across sequences and Fig. 5 shows qualitative rendering results on KITTI road scenes. These results indicate that *MOGS* significantly outperforms monocular-based GS renderers with fewer iterations and lower training cost. Meanwhile, *MOGS* achieves comparable rendering quality to the SOTA costly LiDAR-enabled baseline, demonstrating the practicality of *MOGS*.

C. Ablation Studies

We now evaluate the effectiveness of each individual module of *MOGS*.

Multi-scale Shape Consensus. Removing the multi-scale shape consensus module degrades object coherence on boundaries and small parts, leading to 1.39 dB *PSNR* drop and 0.012 *AbsRel* on average. The quality-steps curves shift right, and we observe more floaters along depth discontinuities, indicating less reliable Gaussian initialization in weak-texture regions.

Cross-object Depth Refinement. Disabling the depth refinement yields the largest performance degradation (-0.78 dB *PSNR*, -0.010 *SSIM*, 0.012 *AbsRel*), confirming that per-object metrization is critical for anchoring LFM depth



Fig. 5: Rendering on KITTI road scenes. Each row shows one representative frame. Columns (left to right) are Ground truth, MonoGS, DepthSplat, GS-LIVM, and MOGS. Our method reconstructs sharper thin structures, preserves depth discontinuities at boundaries, and suppresses long range floaters.

TABLE III: Rendering results on outdoor road sequences KITTI.

Method	Metric	00	01	02	03	04	05	06	07	08	Avg.
MonoGS	PSNR \uparrow	14.5	12.0	14.3	16.1	16.3	15.0	14.2	16.0	13.5	14.7
	SSIM \uparrow	0.492	0.421	0.478	0.543	0.548	0.503	0.472	0.541	0.463	0.495
	LPIPS \downarrow	0.756	0.826	0.758	0.714	0.705	0.742	0.771	0.712	0.782	0.751
DepthSplat	PSNR \uparrow	18.1	16.5	18.1	19.9	20.1	18.7	18.2	19.7	17.8	18.6
	SSIM \uparrow	0.581	0.509	0.571	0.621	0.629	0.589	0.571	0.619	0.560	0.584
	LPIPS \downarrow	0.503	0.582	0.512	0.443	0.432	0.478	0.501	0.442	0.523	0.490
GS-LIVM	PSNR \uparrow	20.4	18.5	20.2	21.3	22.1	20.4	19.4	21.6	19.5	20.4
	SSIM \uparrow	0.640	0.583	0.647	0.706	0.689	0.675	0.645	0.689	0.590	0.652
	LPIPS \downarrow	0.389	0.502	0.381	0.309	0.312	0.363	0.381	0.342	0.443	0.381
Ours	PSNR \uparrow	20.1	18.1	20.1	21.5	21.6	20.6	19.6	21.3	19.0	20.0
	SSIM \uparrow	0.662	0.603	0.651	0.703	0.701	0.671	0.642	0.702	0.612	0.659
	LPIPS \downarrow	0.387	0.503	0.379	0.308	0.309	0.361	0.381	0.339	0.441	0.378

and preserving intra-object shape. Removing this refinement module breaks outlier robustness and introduces a per-mask scale offset that propagates to the Gaussian size and placement. The optimizer then overcompensates with more primitives and steps, yet still converges more slowly and with more artifacts across the scene.

As reported in Table IV, the full MOGS configuration delivers the best quality. Disabling MSC introduces visible floaters near object boundaries and yields a 6.95% decrease in *PSNR* and a 1.97% decrease in *SSIM*, together with a 20.69% increase in *AbsRel*. This suggests that without multi-scale merging and parametric shape fitting, object-level geometric anchors become unreliable, which leads to poor initialization and unstable optimization. When both MSC and CDR are removed, the effect is larger. *PSNR* decreases by 15.90% and *SSIM* by 3.64%, while *AbsRel* increases by 34.48%, underscoring the importance of cross object depth refinement for global consistency and rendering quality.

V. CONCLUSION

This paper presents MOGS, a monocular object-guided pipeline for 3D Gaussian Splatting in large scenes that replaces costly active LiDAR depth with object-anchored, metrized dense depth derived from sparse SfM cues, enabling

TABLE IV: Ablation results on sequence. Higher is better for PSNR/SSIM, lower is better for AbsRel.

Variant	PSNR \uparrow	SSIM \uparrow	AbsRel \downarrow
full MOGS	20.0	0.659	0.058
w/o MSC	18.61	0.646	0.070
w/o MSC + CDR	16.82	0.635	0.078

MSC stands for Multi-scale Shape Consensus;

CDR stands for Cross-object Depth Refinement.

strong Gaussian initialization and high-fidelity rendering. The core insight is to treat objects as carriers of metric geometry. MOGS has two key components: 1) a multi-scale shape consensus module that extracts object-level primitives from image semantics and propagates SfM depth within each object via these primitives; and 2) a cross-object depth refinement module that yields a metrically coherent depth field through a three-term objective, including geometric consistency with LFM depth, LFM prior anchoring, and edge-aware smoothness. Extensive experiments on public datasets show that, using a low-cost VI sensor suite, MOGS reduces training time and memory footprint while achieving rendering quality competitive with costly LiDAR-based approaches.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 52571413, Grant 52031009, Grant 52401423, and Grant 62402353; and in part by the EPSRC with RC under Grant EP/Y027787/1 and UKRI under Grant EP/Y028317/1.

REFERENCES

- [1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139:1–139:14, 2023.
- [2] Y. Duan, F. Wei, Q. Dai, Y. He, W. Chen, and B. Chen, “4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes,” in *ACM SIGGRAPH*, 2024, pp. 1–11.
- [3] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, “4d gaussian splatting for real-time dynamic scene rendering,” in *IEEE/CVF Proc. CVPR*, 2024, pp. 20 310–20 320.
- [4] S. Hong, J. He, X. Zheng, C. Zheng, and S. Shen, “Liv-gaussmap: Lidar-inertial-visual fusion for real-time 3d radiance field map rendering,” *IEEE Robotics and Automation Letters*, 2024.
- [5] C. Zhao, S. Sun, R. Wang, Y. Guo, J.-J. Wan, Z. Huang, X. Huang, Y. V. Chen, and L. Ren, “Tclc-gs: Tightly coupled lidar-camera gaussian splatting for autonomous driving,” in *European Conference on Computer Vision (ECCV)*. Springer, 2024, pp. 91–106.
- [6] X. Lang, L. Li, C. Wu, C. Zhao, L. Liu, Y. Liu, J. Lv, and X. Zuo, “Gaussian-lic: Real-time photo-realistic slam with gaussian splatting and lidar-inertial-camera fusion,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 8500–8507.
- [7] S. Hong, C. Zheng, Y. Shen, C. Li, F. Zhang, T. Qin, and S. Shen, “Gs-livo: Real-time lidar, inertial, and visual multi-sensor fused odometry with gaussian mapping,” *IEEE Transactions on Robotics*, vol. 41, no. 0, pp. 4253–4268, 2025.
- [8] K. Wu, Z. Zhang, M. Tie, Z. Ai, Z. Gan, and W. Ding, “Vings-mono: Visual-inertial gaussian splatting monocular slam in large scenes,” *arXiv preprint arXiv:2501.08286*, 2025.
- [9] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, “Gaussian splatting slam,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 039–18 048.
- [10] Y. Hu, R. Liu, M. Chen, P. Beerel, and A. Feng, “Splatmap: Online dense monocular slam with 3d gaussian splatting,” *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 8, no. 1, pp. 1–18, 2025.
- [11] C. Cheng, S. Yu, Z. Wang, Y. Zhou, and H. Wang, “Outdoor monocular slam with global scale-consistent 3d gaussian pointmaps,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.
- [12] “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” <https://github.com/HKUST-Aerial-Robotics/VINS-Mono>, Online; Accessed: 29 June, 2020.
- [13] S. Cao, X. Lu, and S. Shen, “Gvins: Tightly coupled gnss-visual-inertial fusion for smooth and consistent state estimation,” *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2004–2021, 2022.
- [14] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.
- [15] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, “Depth anything: Unleashing the power of large-scale unlabeled data,” in *IEEE/CVF Proc. CVPR*, 2024, pp. 10 371–10 381.
- [16] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, “Depth anything v2,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 21 875–21 911, 2024.
- [17] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, “Mip-splatting: Alias-free 3d gaussian splatting,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 19 447–19 456.
- [18] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, “Scaffold-gs: Structured 3d gaussians for view-adaptive rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 654–20 664.
- [19] Y. Liu, C. Luo, L. Fan, N. Wang, J. Peng, and Z. Zhang, “Citygaussian: Real-time high-quality large-scale scene rendering with gaussians,” in *European Conference on Computer Vision*. Springer, 2024, pp. 265–282.
- [20] J. Lin, Z. Li, X. Tang, J. Liu, S. Liu, J. Liu, Y. Lu, X. Wu, S. Xu, Y. Yan *et al.*, “Vastgaussian: Vast 3d gaussians for large scene reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5166–5175.
- [21] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, “A vote-and-verify strategy for fast spatial verification in image retrieval,” in *Asian Conference on Computer Vision (ACCV)*, 2016.
- [22] Y. Xie, Z. Huang, J. Wu, and J. Ma, “Gs-livm: Real-time photo-realistic lidar-inertial-visual mapping with gaussian splatting,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 26 869–26 878.
- [23] C. Zhang, Y. Shen, and F. Zhu, “Icp-3dgs: Sfm-free 3d gaussian splatting for large-scale unbounded scenes,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2025.
- [24] Y. Lee, J. Choi, D. Jung, J. Yun, S. Ryu, D. Manocha, and S. Yeon, “Mode-gs: Monocular depth guided anchored 3d gaussian splatting for robust ground-view scene rendering,” *arXiv preprint arXiv:2410.04646*, 2024.
- [25] K. Deng, Y. Zhang, J. Yang, and J. Xie, “Gigaslam: Large-scale monocular slam with hierarchical gaussian splats,” *arXiv preprint arXiv:2503.08071*, 2025.
- [26] L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. Van Gool, and F. Yu, “Unidepth: Universal monocular metric depth estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 106–10 116.
- [27] H. Xu, S. Peng, F. Wang, H. Blum, D. Barath, A. Geiger, and M. Pollefeys, “Depthspat: Connecting gaussian splatting and depth,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16 453–16 463.
- [28] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.
- [29] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, “Usac: A universal framework for random sample consensus,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 2022–2038, 2013.
- [30] C. Fan, S. Zhang, K. Liu, S. Wang, Z. Yang, and W. Wang, “Enhancing mmwave radar point cloud via visual-inertial supervision,” in *IEEE Proc. ICRA*, 2024, pp. 9010–9017.
- [31] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The international journal of robotics research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [32] Y. Liao, J. Xie, and A. Geiger, “Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.
- [33] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, “Depth anything v2,” *Advances in Neural Information Processing Systems*, 2024.
- [34] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun, “Depth pro: Sharp monocular metric depth in less than a second,” *arXiv preprint arXiv:2410.02073*, 2024.
- [35] S. F. Bhat, R. Birkel, D. Wofk, P. Wonka, and M. Müller, “Zoedepth: Zero-shot transfer by combining relative and metric depth,” *arXiv preprint arXiv:2302.12288*, 2023.
- [36] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen, “Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.