electronics

MDPI

Article

Stereo-GS: Online 3D Gaussian Splatting Mapping Using Stereo Depth Estimation

Junkyu Park †, Byeonggwon Lee †, Sanggi Lee and Soohwan Song *

Department of Computer Science and Artificial Intelligence, Dongguk University, Seoul 04620, Republic of Korea; dannypk99@dgu.ac.kr (J.P.); lbg030@dgu.ac.kr (B.L.); sglee@dgu.ac.kr (S.L.)

* Correspondence: songsh@dongguk.edu

† These authors contributed equally to this work.

Abstract

We present Stereo-GS, a real-time system for online 3D Gaussian Splatting (3DGS) that reconstructs photorealistic 3D scenes from streaming stereo pairs. Unlike prior offline 3DGS methods that require dense multi-view input or precomputed depth, Stereo-GS estimates metrically accurate depth maps directly from rectified stereo geometry, enabling progressive, globally consistent reconstruction. The frontend combines a stereo implementation of DROID-SLAM for robust tracking and keyframe selection with FoundationStereo, a generalizable stereo network that needs no scene-specific fine-tuning. A two-stage filtering pipeline improves depth reliability by removing outliers using a variance-based refinement filter followed by a multi-view consistency check. In the backend, we selectively initialize new Gaussians in under-represented regions flagged by low PSNR during rendering and continuously optimize them via differentiable rendering. To maintain global coherence with minimal overhead, we apply a lightweight rigid alignment after periodic bundle adjustment. On EuRoC and TartanAir, Stereo-GS attains state-of-the-art performance, improving average PSNR by 0.22 dB and 2.45 dB over the best baseline, respectively. Together with superior visual quality, these results show that Stereo-GS delivers high-fidelity, geometrically accurate 3D reconstructions suitable for real-time robotics, navigation, and immersive AR/VR applications.

Keywords: 3D Gaussian Splatting; SLAM; stereo depth estimation; online mapping; neural rendering

Academic Editors: Wen-Jing Zhou, Hongbo Zhang and Yuyin Zhou

Received: 24 September 2025
Revised: 7 November 2025
Accepted: 11 November 2025
Published: 14 November 2025

Citation: Park, J.; Lee, B.; Lee, S.; Song, S. Stereo-GS: Online 3D Gaussian Splatting Mapping Using Stereo Depth Estimation . *Electronics* **2025**, *14*, 4436. https://doi.org/10.3390/electronics14224436

1. Introduction

Recent advances in 3D scene reconstruction have enabled photorealistic rendering from sparse inputs, largely driven by neural representations such as Neural Radiance Fields (NeRFs) [1] and their extensions [2,3]. Among these, 3D Gaussian Splatting (3DGS) [4] has emerged as a compelling alternative, offering real-time rendering while maintaining high geometric and visual fidelity. In 3DGS, scenes are represented as collections of translucent 3D Gaussian primitives that are differentiably splatted onto the image plane for rendering. Despite these advantages, most existing 3DGS methods [4–7] are designed for offline operation, relying on precomputed depth maps or dense multi-view observations. This limits their applicability in online Simultaneous Localization and Mapping (SLAM) scenarios, where efficiency, scalability, and robustness under partial observations are critical.

In this paper, we propose Stereo-GS, a framework for online 3DGS modeling from real-time stereo image streams. Unlike monocular 3DGS-based SLAM systems [8–13],

which suffer from scale ambiguity and limited depth precision, our method exploits the geometric constraints of rectified stereo pairs to produce metrically accurate, dense depth maps. This enables direct construction and progressive refinement of a globally consistent 3DGS representation as new stereo frames arrive.

To this end, our system is organized into a frontend and a backend, each optimized for real-time construction and refinement of the 3DGS model. The frontend estimates camera poses and depth, and it comprises two key components: First, the stereo implementation of DROID-SLAM [14] performs robust visual tracking and selects keyframes using dense optical flow across stereo sequences. Second, FoundationStereo [15], a state-of-the-art, highly generalizable stereo matching network, provides accurate depth estimates across diverse environments without scene-specific fine-tuning. To further improve depth reliability, we employ a two-stage filtering pipeline that suppresses outliers via a refinement-variance filter followed by a multi-view consistency check. This ensures that only high-confidence, geometrically consistent depth information is passed to the backend, supporting accurate and stable 3D reconstruction over time.

In the backend, the system incrementally constructs and optimizes the 3DGS model. It selectively initializes new Gaussian primitives in under-represented regions, periodically performs global bundle adjustment (GBA), and continuously refines Gaussian parameters via differentiable rendering. To reduce computational overhead, we introduce a lightweight rigid map-alignment step that aligns the global map with updated camera poses after each GBA, avoiding full remapping while maintaining model consistency. As shown in Figure 1, the synergy of stereo geometry, filtering, and real-time optimization enables robust and accurate online 3D reconstruction even in unconstrained, challenging environments.

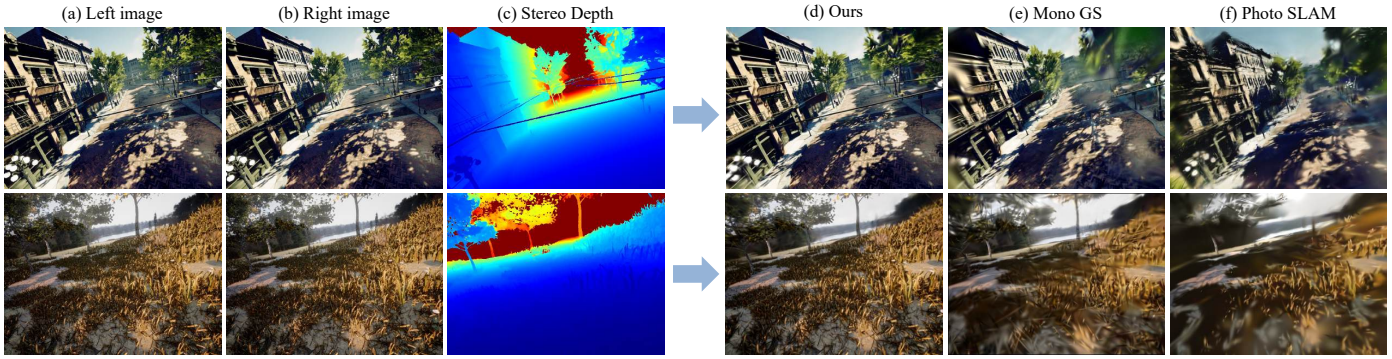


Figure 1. Our method takes a left-right image pair as the input, estimates a precise depth map based on stereo vision, and utilizes it to generate a high-quality 3DGS model. In the depth map, blue represents closer regions and red represents more distant regions. Our method produces cleaner rendering results compared to existing methods, MonoGS [10] and PhotoSLAM [9].

The main contributions of this work are as follows:

- We propose a framework for online, high-fidelity 3DGS reconstruction driven by stereo depth estimation. Accurate, dense Gaussian primitives are initialized directly from stereo-derived depth maps.
- We introduce a robust stereo-based depth estimation pipeline with a comprehensive two-stage filtering mechanism that sequentially estimates depths from incoming frames, integrates them for temporal consistency, and effectively removes outliers.
- We provide a thorough evaluation on challenging benchmarks [16,17], demonstrating state-of-the-art performance. Our method achieves an average PSNR of 23.70 on EuRoC [16] and 22.47 on TartanAir [17], outperforming existing online 3DGS-SLAM methods [9,10] with sharper details and fewer artifacts.

2. Related Work

2.1. Stereo SLAM

Classical stereo visual SLAM systems exploit the fixed baseline of a calibrated camera pair to recover metric scale and mitigate scale drift. Prominent feature-based pipelines such as ORB-SLAM [18,19] and VINS-Fusion [20] perform keypoint detection, stereo matching, and bundle adjustment on a local keyframe graph, achieving reliable and efficient tracking across diverse environments. Direct and semi-dense methods like DSO [21] and SVO2 [22] rely on photometric alignment to improve accuracy in low-texture regions, but remain sensitive to illumination changes and require careful photometric modeling. Although these approaches estimate camera poses robustly, their reconstructions are typically sparse or semi-dense; dense scene models generally require additional modules such as TSDF fusion [23] or voxel-based fusion pipelines [24].

Integrating a stereo depth estimation module into a SLAM pipeline enables dense depth prediction for dense 3D reconstruction. Recently, learning-based stereo matching has substantially improved both quality and generalization. Early cost-volume Convolutional Neural Network (CNN) methods (PSMNet [25], GA-Net [26]) have been followed by transformer and recurrent architectures (RAFT-Stereo [27] and IGEV-Stereo [28]), which deliver sharper disparity estimates and stronger performance in challenging regions such as occlusions and thin structures. More recently, foundation-style stereo models [15,29] have emerged, combining strong monocular priors (e.g., DepthAnything [30]) with hybrid cost aggregation. This design enables effective cross-domain generalization without scene-specific fine-tuning. In particular, FoundationStereo [15] achieves robust, highly accurate depth on in-the-wild imagery, demonstrating state-of-the-art zero-shot generalization.

DROID-SLAM [14] introduces dense, learned correspondence fields and jointly optimizes camera poses and depth. Its architecture, adapted for stereo inputs, leverages dense matches within a differentiable optimization framework, improving robustness under fast motion or repetitive textures. However, when paired with conventional mapping backends, maintaining real-time throughput often necessitates low-resolution depth maps, which in turn yield sparse or over-smoothed scene models.

Our Stereo-GS couples a stereo-adapted DROID-SLAM [14] frontend with Foundation-Stereo [15] to recover metrically accurate, dense depth maps online. We further stabilize these estimates via a two-stage filtering pipeline, refinement-variance filtering followed by multi-view consistency checks, which mitigates far-depth noise and enables reliable, dense online reconstruction.

2.2. Differentiable Rendering SLAM

Neural, differentiable scene representations have reshaped dense SLAM. NeRF-based systems [31–34] jointly optimize camera poses and volumetric radiance fields from monocular or RGB-D streams, producing photorealistic renderings with dense geometry. Despite these strengths, NeRF pipelines often incur substantial compute and memory costs and exhibit latencies that hinder strict real-time operation.

3DGS [4] offers a compelling alternative to NeRF [1] by representing scenes as sets of translucent, anisotropic Gaussians rendered via differentiable splatting. Compared with NeRF, 3DGS trains significantly faster and supports real-time rendering while maintaining competitive fidelity. Building on this, several works have brought 3DGS into SLAM. SplaTAM [8] integrates a dense photometric tracking objective with an online-updated Gaussian map, enabling end-to-end differentiable pose–map co-optimization. MonoGS [10] realizes fully monocular 3DGS-SLAM by incrementally initializing and refining Gaussians from a single video stream, using differentiable reprojection and regularization to stabilize map growth. PhotoSLAM [9] further emphasizes photometric consistency on the Gaussian

representation, coupling view-synthesis losses with geometric constraints to better align appearance and geometry during online mapping. MVS-GS [12,13] extends this line by employing an online multi-view stereo (MVS) frontend to produce dense depth and point clouds that seed and update Gaussians.

Despite strong results, most monocular variants [8–13] inherit scale drift and limited depth precision, whereas RGB-D extensions [35,36] rely on active sensors with restricted range and characteristic noise. These limitations motivate stereo formulations that recover metric scale and dense depth purely from passive cameras while preserving the efficiency advantages of differentiable splatting. Although MonoGS [10] and PhotoSLAM [9] provide stereo-compatible modes, their pipelines remain largely monocular-centric and thus do not fully exploit stereo geometry to extract high-accuracy, dense metric depth.

In contrast, Stereo-GS unifies a stereo-adapted DROID-SLAM [14] frontend with FoundationStereo [15] to obtain dense, metric-scale depth. We also maintain global consistency with periodic bundle adjustment and lightweight rigid map alignment. The result is an end-to-end stereo system that preserves the speed and fidelity of 3DGS while achieving reliable, real-time dense reconstruction from streaming stereo inputs.

3. Proposed Method

This study presents an online 3D reconstruction framework that incrementally builds a high-fidelity 3DGS [4] representation from a continuous stereo image stream. Each stereo pair, composed of rectified left and right images $(I^{\text{left}}, I^{\text{right}})$, enables the estimation of a disparity map $\tilde{D}$, which encodes horizontal pixel displacements between corresponding points in the stereo images. This disparity map is subsequently transformed into a depth map D, providing scene geometry at metric scale. By leveraging a sequence of depth maps $\{D_k\}$ derived from incoming stereo frames, we propose an efficient and accurate online method for progressively generating detailed 3DGS models suitable for photorealistic rendering and real-time applications.

3.1. System Overview

Figure 2 illustrates the overall architecture of the proposed online 3DGS modeling system. Our approach adopts the MVS-based pipeline introduced in MVS-GS [12], which consists of two key components: a frontend and a backend. The frontend is responsible for estimating both camera poses and depth maps from incoming stereo keyframes, while the backend incrementally generates and updates 3D Gaussian splats for the scene representation within the 3DGS framework. These two modules operate asynchronously on separate threads, enabling real-time and efficient reconstruction.

Specifically, the frontend estimates camera trajectories from the stereo image stream using the stereo implementation of DROID-SLAM [14], which utilizes dense optical flow to establish fine-grained pixel correspondences across frames. The stereo implementation extends its monocular counterpart by retaining the core recurrent optimization framework while adding dense spatial constraints derived from the left-right image pair. These spatial constraints, manifested as simultaneously estimated disparity map, are integrated with the known metric baseline and integrated as a strong intra-stereo co-visibility factor into the factor graph. This process anchors the system to a real-world scale and eliminates the scale ambiguity inherent in the monocular version.

For each stereo pair $(I_k^{\text{left}}, I_k^{\text{right}})$, the left image is used as the reference view for tracking the camera pose T_k. Owing to the fixed extrinsic calibration between the stereo cameras, the pose of the right image can be directly inferred from the estimated pose of the left image.

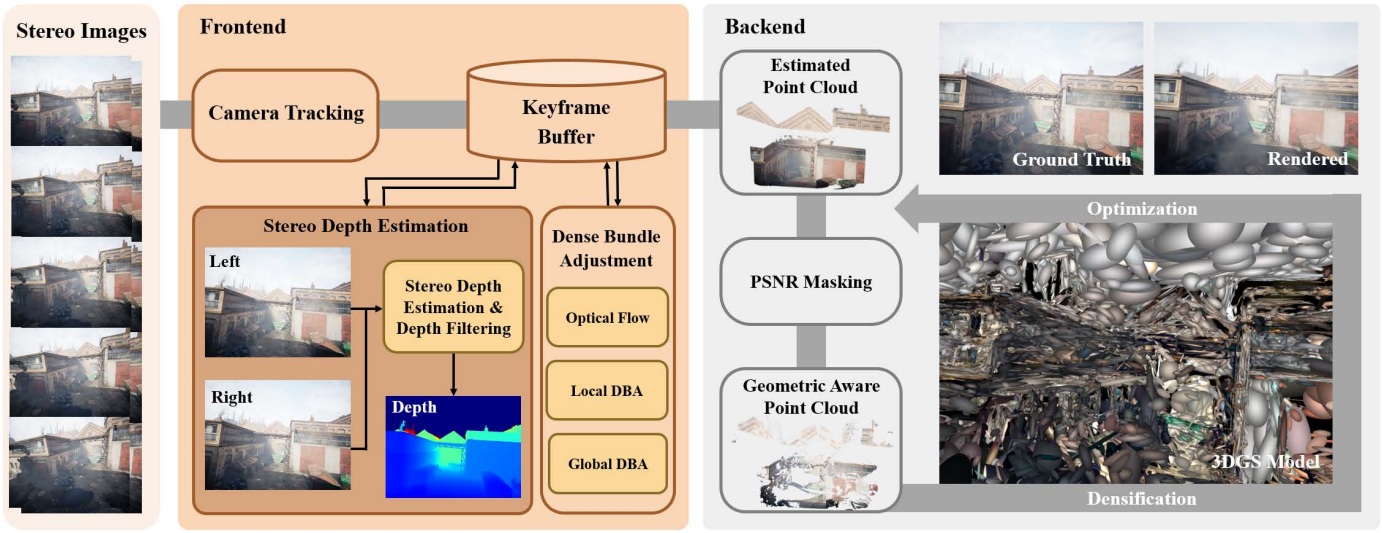


Figure 2. System overview. Our system adopts a parallel frontend–backend architecture. The frontend performs two key functions: it first estimates keyframe poses with DROID-SLAM [14], and then it predicts and refines a depth map for each keyframe using our novel stereo depth estimation with depth filtering block. This component, introduced in this work, leverages FoundationStereo [15] and a dedicated depth-outlier filter. The backend then uses these refined depths to initialize new Gaussian primitives, integrate them into the 3DGS scene, and continuously optimize the model thereafter. In the depth map, blue represents closer regions and red represents more distant regions.

As the stereo sequence progresses, the frontend incrementally constructs a keyframe graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where each node $F_k \in \mathcal{N}$ corresponds to an extracted keyframe. Each keyframe F_k comprises an RGB image I_k, a camera pose T_k, and an estimated depth map D_k. Each edge $(F_k, F_l) \in \mathcal{E}$ represents pairwise co-visibility relationships, determined by both temporal adjacency and sufficient stereo baseline overlap. To improve local pose accuracy, dense bundle adjustment (DBA) [14] is performed over neighboring keyframes. Furthermore, to suppress long-term drift and ensure global consistency of the trajectory and scene structure, global DBA is periodically applied every 30 keyframes, unlike the original approach in [14], where global DBA was invoked only after the entire mapping process had concluded. This modification enables more robust online optimization. In addition, intra-stereo co-visibility is explicitly incorporated to introduce auxiliary scale constraints within each stereo pair. This stereo-based formulation inherently supports the recovery of absolute scale, in contrast to monocular SLAM systems [18], thereby enhancing the metric accuracy of the reconstructed 3D scene.

For per-keyframe depth estimation, we adopt the FoundationStereo network [15], a learning-based stereo matching model known for its strong generalization across diverse and unseen environments. Unlike conventional stereo approaches that often overfit to domain-specific training data, FoundationStereo consistently delivers accurate depth predictions even in untrained scenes. To further improve the reliability of depth estimation, we incorporate a dedicated outlier rejection module that filters out low-confidence or erroneous depth values. This refinement process considers multiple error sources, including stereo mismatches, uncertainty arising from disparity-to-depth conversion, and occlusions where pixels are visible in only one of the two views. As a result, the frontend outputs refined depth maps $\hat{D}_k$, which provide a robust and accurate foundation for the subsequent stages of 3D reconstruction.

The backend module receives, for each keyframe F_k, the estimated camera pose T_k and the corresponding refined depth map $\hat{D}_k$ generated by the frontend. It incrementally updates the parameters of the 3DGS model through a continuous optimization

pipeline. Initially, the model is seeded using a selected subset of the earliest keyframes. As new keyframes arrive, additional Gaussians are dynamically introduced into the scene representation.

To maintain model compactness and reduce redundancy, the backend detects uncovered regions in the current depth map that are inadequately represented by the existing Gaussians. Only these regions are converted into 3D point clouds. This selective conversion strategy ensures both precision and efficiency, as it avoids redundant reconstruction of already-modeled areas. The newly generated 3D points serve as initial seeds for creating new Gaussian primitives, which are subsequently integrated into the global model. The backend then performs joint optimization of all Gaussian parameters to refine their positions, shapes, opacities, and colors, ensuring high-fidelity rendering.

3.2. Stereo Depth Estimation

We employ FoundationStereo [15] to estimate dense depth maps from stereo image pairs associated with each keyframe. This model exhibits strong zero-shot generalization across diverse and previously unseen scenes, eliminating the need for scene-specific fine-tuning. Its robustness stems from the integration of monocular priors derived from DepthAnything [30] with a hybrid cost volume aggregation framework, which incorporates axial-planar convolution layers and a disparity transformer module.

For each keyframe F_k, the left image I_k^{left} is used as the reference view, while the right image I_k^{right} serves as the source view. FoundationStereo first computes a disparity map $\tilde{D}_k$ for the reference image. This disparity is then converted into a metric depth map D_k using the intrinsic camera parameters, the focal length f and stereo baseline B, according to the following equation:

$$D_k(x) = \frac{f \cdot B}{\tilde{D}_k(x)} \tag{1}$$

where $D_k(x)$ and $\tilde{D}_k(x)$ denote the depth and disparity at pixel x, respectively.

This stereo estimation framework enables real-time generation of geometrically accurate and dense depth maps with high reliability. In contrast to traditional stereo learning models [27,28], FoundationStereo demonstrates superior resilience under challenging conditions, including textureless surfaces, occlusions, and extreme illumination variations. These properties make it particularly well-suited as a backbone for online 3D mapping in complex and unconstrained environments.

3.3. Depth Outlier Filtering

To enhance the accuracy and stability of depth estimation within the frontend, we propose a two-stage filtering scheme comprising (i) refinement variance-based filtering and (ii) multi-view consistency filtering. We first remove far-depth outliers by thresholding pixels with very small disparities. Next, we apply refinement variance-based filtering. FoundationStereo leverages a strong monocular prior and refines disparity via ConvGRU-based residual updates [27]; we exploit this iterative process to model per-pixel uncertainty. Let $\mathbf{D}^i$ denote the disparity map predicted at the i-th refinement step. We compute the per-pixel variance over N iterations,

$$\sigma^2(x) = \frac{1}{N-1} \sum_{i=1}^{N} \left(\mathbf{D}^i(x) - \bar{\mathbf{D}}(x) \right)^2 \tag{2}$$

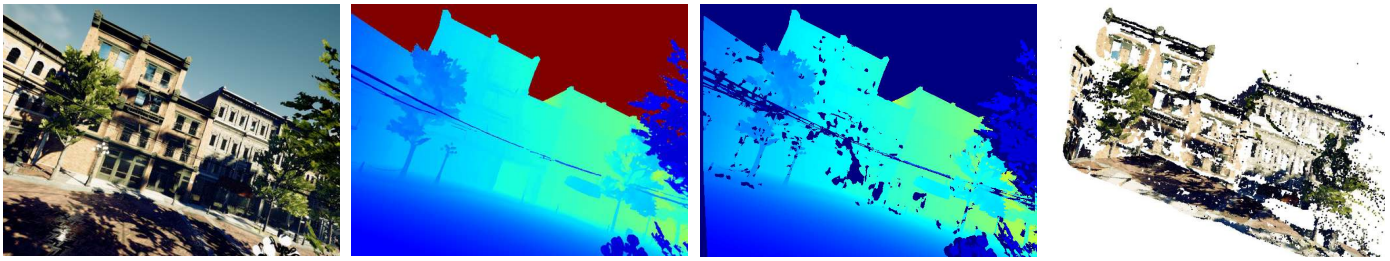
where $\bar{\mathbf{D}}$ is the iteration-wise mean disparity. A high variance indicates that the estimate failed to converge and is therefore low-confidence, whereas a low variance implies a stable, reliable estimate. We treat the variance map as an inverse-confidence measure and apply quantile thresholding, excluding the top 5% highest-variance pixels from the final depth

map. We empirically set this threshold to offer a good balance between outlier removal and map completeness; it effectively eliminates most unstable predictions without being overly aggressive and creating large holes in the depth map.

Next, following standard MVS practice [37,38], we enforce geometric consistency over a short temporal window of the five preceding frames. Using the estimated relative poses and camera intrinsics, each prior depth map $\mathbf{D}_k$ (for $k \in \{t-1, \ldots, t-5\}$) is unprojected to 3D and reprojected into the current frame t to obtain $\mathbf{D}_{k\to t}^{\text{proj}}$. For every pixel that remains visible after z-buffer and occlusion checks, we compare the reprojected depth with the current estimate $\mathbf{D}_t$ and compute a normalized discrepancy

$$\delta_{k\to t}(x) = \frac{|\mathbf{D}_{k\to t}^{\text{proj}}(x) - \mathbf{D}_t(x)|}{\mathbf{D}_t(x)} \tag{3}$$

A pixel is accepted if it receives at least $m = 3$ inlier votes among $K = 5$ frames with $\delta_{k\to t}(x) < \tau$; otherwise it is marked geometrically inconsistent and discarded. The threshold τ governs the accuracy-completeness trade-off: tighter values remove more outliers from occlusions, motion blur, or far-depth noise, whereas looser values preserve coverage in texture-poor regions. This multi-view voting scheme yields a cleaner and more stable depth field prior to downstream mapping. Figure 3 illustrates the depth-filtering pipeline: the initial stereo depth map is refined via outlier removal to produce a filtered map, which is then used to generate the final 3D point cloud.



(a) Color Image (b) Stereo Depth Map (c) Filtered Depth Map (d) Point Cloud

Figure 3. An illustration of (**a**) the input color image, (**b**) the depth map obtained via stereo matching, (**c**) the filtered depth map, and (**d**) the resulting point cloud. In the depth map, blue represents closer regions and red represents more distant regions.

3.4. Online 3DGS Mapping

Our system performs online 3D scene reconstruction by constructing a point cloud P_k for each keyframe F_k, using the filtered depth map $\hat{D}_k$ and its corresponding camera pose T_k. These point clouds form the basis for initializing Gaussian primitives in the 3DGS representation. In the early stage, the initial set of Gaussians is instantiated from the point clouds derived from the first few keyframes.

As new keyframes arrive, the 3DGS model is incrementally expanded through the incorporation of additional Gaussians. These new Gaussians are generated from depth maps $\{\hat{D}_k\}$ of incoming frames, but only in regions that remain under-reconstructed or exhibit low rendering fidelity. To efficiently identify such areas and avoid unnecessary computation or uncontrolled Gaussian growth, we adopt the strategy proposed in [12], which leverages view-based quality metrics. Specifically, we compute the Peak Signal-to-Noise Ratio (PSNR) between rendered views and their corresponding input images. PSNR is an image quality metric that expresses how close a reconstructed image is to a reference by comparing the peak signal level to the reconstruction error. Regions with PSNR values below a predefined threshold are flagged as unexplored and prioritized for additional Gaussian generation.

To maintain global consistency and improve the accuracy of both geometry and appearance, we periodically perform global DBA [14] alongside joint Gaussian parameter optimization. Since the entire 3D Gaussian map is built incrementally in the camera coordinate frames, the reconstructed scene must be aligned with the refined camera trajectories resulting from global DBA. For this purpose, we apply a rigid transformation to the entire 3D map after each global DBA iteration, effectively re-aligning the Gaussians with minimal computational overhead [13]. This rigid alignment step precedes each subsequent Gaussian optimization phase, ensuring that all Gaussians remain spatially consistent with the most up-to-date camera poses. Thanks to the high-fidelity depth maps generated by the stereo reconstruction pipeline, this rigid transformation is both accurate and efficient, enabling real-time performance without compromising the geometric integrity of the map.

3.5. 3DGS Optimization

We adopt the 3DGS optimization framework proposed by [4], which represents a scene as a collection of transparent 3D Gaussian primitives $\{g_i\}$. Each Gaussian g_i is parameterized by its center position $\mu_i \in \mathbb{R}^3$, covariance matrix $\Sigma_i \in \mathbb{R}^{3\times3}$, opacity $\mathbf{o}_i \in [0,1]$, and color coefficient $\mathbf{c}_i$. To capture view-dependent appearance, the color is modeled using low-order spherical harmonics.

Rendering is performed via a differentiable splatting technique, where each Gaussian is projected onto the image plane and contributes to the final pixel color through front-to-back alpha compositing. The contribution of each Gaussian is modulated by its visibility-weighted influence:

$$\alpha_i = o_i \, g_i(x) \tag{4}$$

where $g_i(x)$ is the probability density of the Gaussian projected to pixel x. The accumulated pixel color $\hat{C}(x)$ is then computed as:

$$\hat{C}(x) = \sum_{i \in M} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{5}$$

This formulation ensures that closer and more opaque Gaussians have a greater impact on the final rendered image.

The backend module continuously refines the parameters of the Gaussians through gradient-based optimization, leveraging differentiable rendering. The optimization updates the Gaussians' spatial positions μ_i, covariances Σ_i, opacities o_i, and color coefficients c_i. The training loss function is defined as a weighted combination of two losses:

$$L = \lambda_{L1} \, L_{L1} + \lambda_{SSIM} \, L_{SSIM} \tag{6}$$

where L_{L1} is a photometric L1 loss measuring the pixel-wise difference between rendered and ground-truth images; $L_{\mathbf{SSIM}}$ is a Structural Similarity Index (SSIM) loss designed to preserve the perceptual quality and structural integrity of the images.

In the stereo setting, we additionally enforce stereo consistency by utilizing the known fixed baseline between the left and right cameras. The scene is rendered from both the left-camera pose and the corresponding right-camera pose. Independent photometric losses are computed for each view and averaged:

$$L^* = \frac{1}{2}(L^{\text{left}} + L^{\text{right}}) \tag{7}$$

This encourages the model to learn appearance features that are consistent across viewpoints while respecting the geometric constraints imposed by stereo vision, thereby improving both photometric robustness and depth accuracy.

4. Experimental Results

4.1. Experimental Setup

To assess the effectiveness of our online 3D modeling framework for neural rendering, we conducted comparative experiments on the TartanAir [17] and EuRoC [16] datasets, covering a diverse set of indoor and outdoor trajectories. For baselines, we used stereo-enabled variants of PhotoSLAM [9] and MonoGS [10], both configured to operate in the same online setting. All methods processed each stereo sequence sequentially without access to future frames and without scene-specific fine-tuning, and they shared the same keyframe schedule produced by the frontend to ensure a fair comparison. Evaluation followed standard image-space criteria: for each processed frame, we rendered views from the current 3D Gaussian map and computed PSNR, SSIM [39], and LPIPS [40] with respect to the corresponding ground-truth images; scores were then averaged per sequence and across sequences. The experiments were run on a desktop with an Intel i9-13900KS CPU and an NVIDIA GeForce RTX 3090 Ti GPU. The models were implemented in PyTorch (Version 2.7.1). CUDA accelerated splatting, rasterization, and gradient accumulation to satisfy real-time constraints.

4.2. Evaluation on EuRoC

Table 1 reports quantitative rendering results on the EuRoC dataset, which consists of stereo sequences from a micro aerial vehicle. Our method achieves the best overall performance across all trajectories, with significant gains in PSNR and SSIM, alongside consistently competitive or improved LPIPS scores. This superior performance demonstrates that our stereo-driven 3DGS pipeline effectively addresses key challenges in stereo vision, such as view inconsistency, occlusions, and far-depth noise, leading to more stable and photometrically accurate reconstructions.

Table 1. Quantitative rendering performance on the EuRoC dataset. The upward arrows (↑) indicate that higher metric values are better, while downward arrows (↓) indicate that lower values are better. The best results for each metric are highlighted in bold.

Method	Metric	MH01	MH02	MH03	MH04	MH05	V1_01	V2_01	Avg.
MonoGS	PSNR↑	**25.88**	17.26	19.59	**25.23**	**24.67**	**28.07**	23.65	23.48
	SSIM↑	**0.85**	0.68	0.71	0.85	0.84	**0.90**	0.83	0.81
	LPIPS↓	**0.17**	0.43	0.38	0.24	0.26	**0.19**	0.28	0.26
PhotoSLAM	PSNR↑	21.23	22.10	20.92	20.22	19.73	23.13	21.95	21.23
	SSIM↑	0.70	0.73	0.70	0.74	0.72	0.78	0.78	0.74
	LPIPS↓	0.30	0.29	0.34	0.34	0.38	0.28	0.30	0.32
Ours	PSNR↑	22.63	**23.34**	**23.83**	23.34	23.12	25.01	**24.12**	**23.70**
	SSIM↑	0.77	**0.79**	**0.81**	**0.88**	**0.86**	0.86	**0.85**	**0.83**
	LPIPS↓	0.27	**0.26**	**0.27**	**0.19**	**0.22**	0.22	**0.23**	**0.24**

Qualitative results in Figure 4 further validate these findings. Our renderings preserve fine structural details like thin edges and high-frequency textures. Our results also suppress artifacts such as ghosting near depth discontinuities and maintain coherence across viewpoints. Together, the quantitative and qualitative evidence establishes a new state-of-the-art in stereo vision-based online 3D reconstruction on the EuRoC benchmark.

4.3. Evaluation on TartanAir

Table 2 summarizes results on TartanAir, a challenging benchmark featuring photorealistic scenes with large viewpoint changes, rapid 6-DoF motion, strong parallax, and challenging lighting. All methods were run in the same online setting with sequential processing and no scene-specific fine-tuning. Across all sequences, our approach delivers

state-of-the-art neural rendering performance, achieving the highest PSNR/SSIM and consistently competitive or lower LPIPS than the baselines. These gains are most pronounced in fast-motion segments and large-baseline transitions. We attribute the improvements to (i) metrically accurate, dense stereo depths from the FoundationStereo frontend, (ii) our two-stage reliability filtering, and (iii) quality-aware Gaussian insertion that targets under-represented regions without uncontrolled map growth.

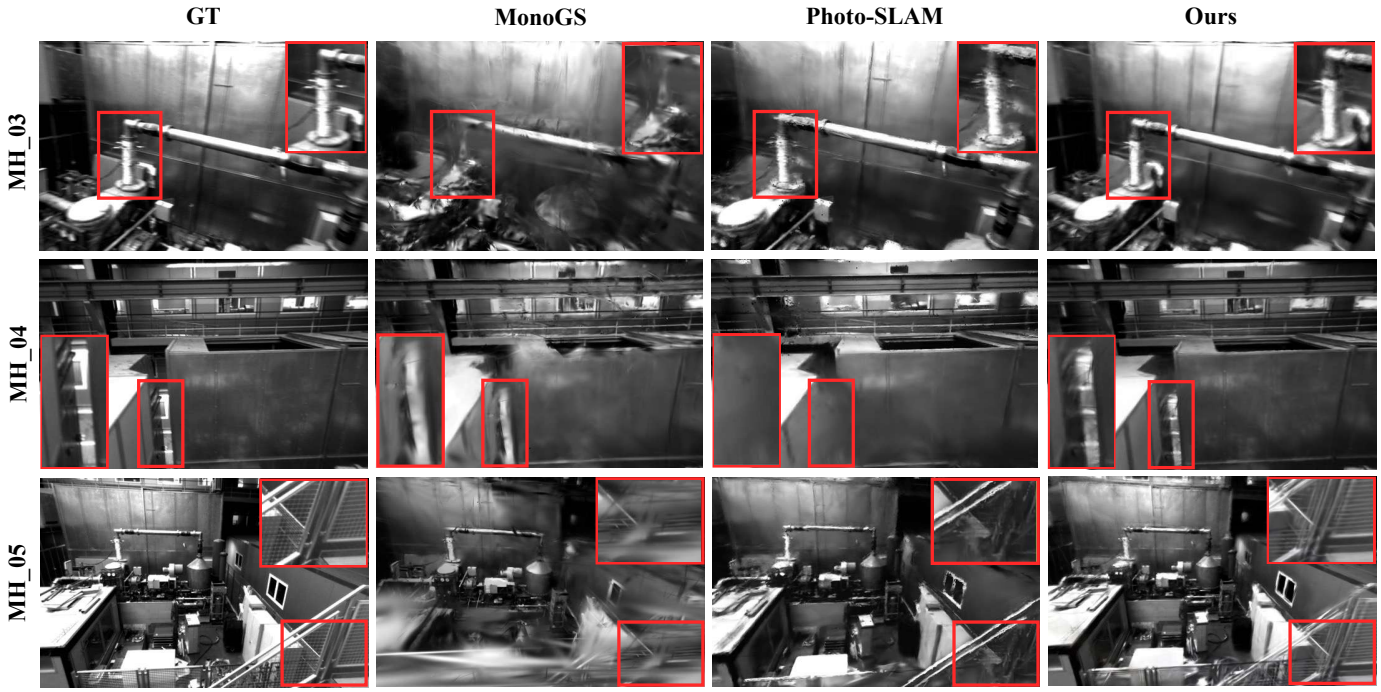


Figure 4. A qualitative evaluation on the EuRoC dataset. Rendering results from the MH_03, MH_04, MH_05 scenes, comparing MonoGS [10], PhotoSLAM [9], and our proposed method. Red boxes are used to highlight regions where our method demonstrates superior rendering quality, such as clearer details and fewer artifacts, compared to other approaches.

Table 2. A quantitative evaluation on the TartanAir dataset. Rendering results comparing MonoGS [10], PhotoSLAM [9], and our method on four challenging scenes. The upward arrows (↑) indicate that higher metric values are better, while downward arrows (↓) indicate that lower values are better. The best results for each metric are highlighted in bold.

Method	Metric	SE000	SE001	SE002	SE003	SE004	SE005	SE006	SE007	Avg.
MonoGS	PSNR↑	20.95	17.76	18.81	17.59	27.92	17.26	16.45	23.38	20.02
	SSIM↑	0.59	0.35	**0.48**	0.65	0.78	0.43	**0.28**	0.64	0.53
	LPIPS↓	0.60	0.69	**0.58**	0.51	**0.55**	0.66	0.76	0.56	0.61
PhotoSLAM	PSNR↑	20.70	16.41	16.95	23.16	25.98	16.28	**17.73**	24.07	19.67
	SSIM↑	0.58	0.30	0.39	0.72	0.81	0.40	0.33	**0.75**	0.50
	LPIPS↓	0.55	0.68	0.73	0.38	0.56	0.64	0.66	**0.33**	0.61
Ours	PSNR↑	**23.49**	**20.32**	**20.64**	**23.5**	**29.28**	**20.00**	17.39	**25.12**	**22.47**
	SSIM↑	**0.67**	**0.58**	0.46	**0.82**	**0.93**	**0.53**	0.34	0.68	**0.61**
	LPIPS↓	**0.48**	**0.45**	0.69	**0.32**	0.59	**0.56**	**0.65**	0.49	**0.53**

Qualitative comparisons in Figure 5 corroborate the metrics: competing methods exhibit ghosting, bleeding near depth discontinuities, and texture blurring under rapid viewpoint changes, whereas our renderings preserve thin structures and high-frequency textures, suppress flicker and reprojection artifacts, and maintain appearance coherence across views. The resulting reconstructions show fewer holes and sharper geometry in

reflective, low-texture, and heavily occluded areas. Taken together, the quantitative and qualitative evidence establishes our method as the strongest among the compared baselines on TartanAir, demonstrating robust generalization to highly dynamic, stereo vision–based online 3D reconstruction.

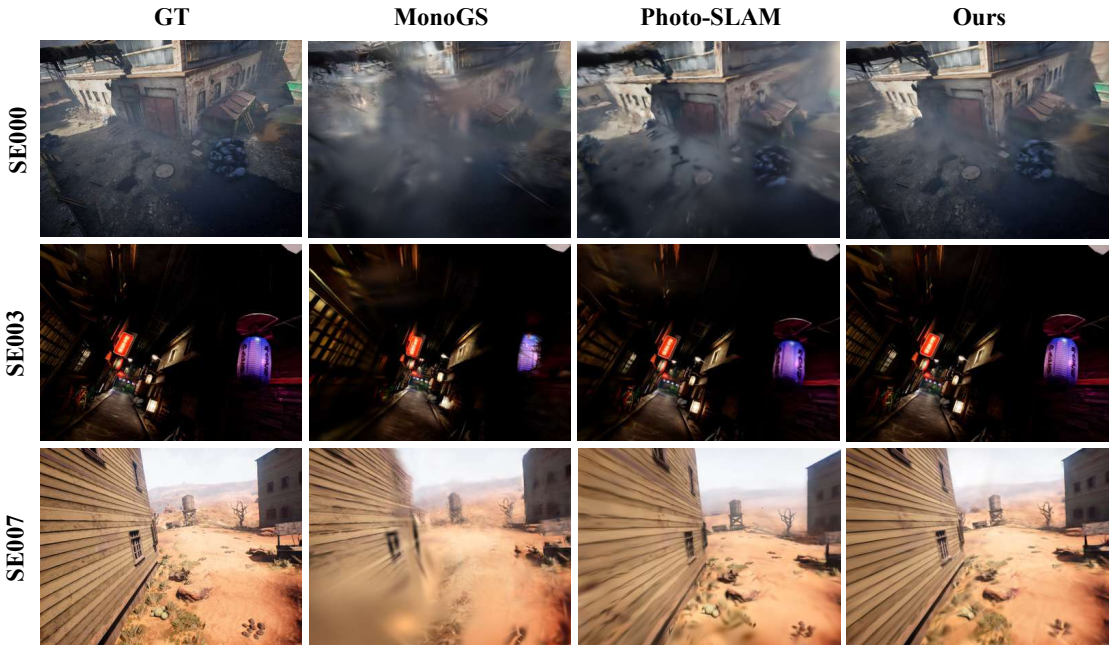


Figure 5. A qualitative evaluation on the TartanAir dataset. Rendering results comparing MonoGS [10], PhotoSLAM [9], and our method on three challenging scenes.

4.4. Ablation Study

We performed an ablation study on the SE000 sequence from the TartanAir dataset to validate the effectiveness of our method's key components. We assessed model variants by selectively enabling or disabling modules, such as: the Stereo Depth component (using FoundationStereo depth or DROID-SLAM's low-resolution depth), Global Bundle Adjustment (GBA), and the mask-based initialization of Gaussians. Each variant was measured for its photometric metrics, number of Gaussians, and frames per second (FPS).

Table 3 summarizes the four variants: Model-A attains the highest FPS but yields the smallest Gaussian set and the weakest rendering fidelity. Model-C delivers the best visual quality while operating at the lowest FPS. Model-D accepts a marginal quality drop relative to Model-C in exchange for a clear FPS gain. Together, these results indicate that initializing Gaussians exclusively in previously unexplored regions is an effective strategy.

Table 3. The comparison results of our method with different model variants on the sequence SE000 of TartanAir dataset. The upward arrows (↑) indicate that higher metric values are better, while downward arrows (↓) indicate that lower values are better. Model variants are defined as follows: A (baseline without key components), B (with Stereo Depth), C (B + Global Bundle Adjustment), D (C + Mask Update).

Model	Stereo Depth	GBA	Mask Update	PSNR↑	SSIM↑	LPIPS↓	#Gaussians↓	FPS↑
A	X	X	X	12.21	0.44	0.95	182	6.91
B	O	X	X	21.77	0.61	0.52	430	5.33
C	O	O	X	23.62	0.67	0.45	402	3.48
D	O	O	O	23.49	0.67	0.48	331	4.19

5. Limitations and Discussion

The proposed Stereo-GS framework has demonstrated superior rendering quality on diverse benchmark datasets, including EuRoC and TartanAir. Notably, by leveraging accurate, metric-scale stereo depth estimation from FoundationStereo and a robust two-stage filtering pipeline, our method maintains high geometric fidelity even in challenging areas. As illustrated in Figures 4 and 5, prior methods like MonoGS and PhotoSLAM often exhibit ghosting and other artifacts in regions with rapid motion or sparse textures. In contrast, Stereo-GS consistently produces sharp and stable 3DGS models, validating the efficacy of our proposed pipeline.

Despite these achievements, several limitations remain. First, our method was predominantly validated in relatively contained environments. When scaling to large-scale outdoor scenes, such as the KITTI dataset [41], the number of 3D Gaussians would increase significantly, leading to substantial storage requirements and computational overhead. Consequently, effective 3DGS map management, including techniques for pruning redundant Gaussians, merging similar primitives, or compressing map data, is essential for long-term SLAM and represents a critical area for future research.

Second, the frontend poses a computational bottleneck. Stereo-GS employs a two-module approach, using DROID-SLAM for pose tracking and FoundationStereo for depth estimation. While running these networks in parallel ensures high accuracy, this approach incurs a significant computational load, particularly for embedded systems in robotics or AR. Future work could explore a lightweight, tightly-coupled frontend architecture, potentially by sharing a common feature extractor, to improve real-time performance and efficiency.

Finally, while our system incorporates periodic GBA to maintain global consistency, the reconstruction quality remains sensitive to severe tracking failures. In extremely challenging scenarios, such as scenes dominated by repetitive textures or rapid motion, even a robust frontend like DROID-SLAM can produce pose errors. These errors can propagate into the 3DGS map, causing the misplacement of Gaussian primitives. To address this, future work should explore the effective use of local bundle adjustment, followed by a corresponding update of the Gaussian points to maintain map consistency.

6. Conclusions

We presented Stereo-GS, a real-time online framework for high-fidelity 3DGS reconstruction from stereo images. Our method overcomes the offline limitations of most 3DGS systems by leveraging stereo geometry for metric-scale depth estimation, eliminating the scale ambiguity inherent in monocular approaches. The core of Stereo-GS is a robust pipeline that integrates a stereo-adapted DROID-SLAM frontend for pose tracking with the FoundationStereo network for dense depth prediction, enhanced by a two-stage filtering mechanism for outlier removal. This reliable geometric input enables the backend to incrementally build and optimize a consistent 3DGS map efficiently.

Experimental results on the EuRoC and TartanAir benchmarks quantitatively demonstrate state-of-the-art performance. Our method's superiority is particularly evident on the challenging TartanAir dataset, where Stereo-GS achieved an average PSNR of 22.47, significantly outperforming the baseline methods by a margin of over 2.45 dB. Our method also secured top performance on the EuRoC dataset with an average PSNR of 23.70. These results confirm our approach effectively handles difficult scenarios, such as fast motion and textureless regions, producing sharper details and fewer artifacts than competing methods.

Author Contributions: Conceptualization, S.S.; Methodology, J.P., Software, B.L.; Formal analysis, S.L. All authors have read and agreed to the published version of this manuscript.

Funding: This work was supported in part by the Korea Institute of Energy Technology Evaluation and Planning (KETEP); the Ministry of Trade, Industry and Energy (MOTIE), Korea (No. 20224000000020); and by the Institute of Information & Communication Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2025-RS-2023-00254592) grant funded by the Korea government (MSIT).

Data Availability Statement: The data are available upon request due to restrictions, e.g., privacy or ethical reasons. The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **2021**, *65*, 99–106. [CrossRef]
2. Müller, T.; Evans, A.; Schied, C.; Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph. (TOG)* **2022**, *41*, 102. [CrossRef]
3. Barron, J.T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; Srinivasan, P.P. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 5855–5864.
4. Kerbl, B.; Kopanas, G.; Leimkühler, T.; Drettakis, G. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Trans. Graph* **2023**, *42*, 139. [CrossRef]
5. Yu, Z.; Chen, A.; Huang, B.; Sattler, T.; Geiger, A. Mip-Splatting: Alias-free 3D Gaussian Splatting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024.
6. Chen, Y.; Xu, H.; Zheng, C.; Zhuang, B.; Pollefeys, M.; Geiger, A.; Cham, T.J.; Cai, J. MVSplat: Efficient 3D Gaussian Splatting from sparse multi-view images. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2024.
7. Bao, K.; Wu, W.; Hao, Y. Gaussian Splatting-Based Color and Shape Deformation Fields for Dynamic Scene Reconstruction. *Electronics* **2025**, *14*, 2347. [CrossRef]
8. Sandström, E.; Zhang, G.; Tateno, K.; Oechsle, M.; Niemeyer, M.; Zhang, Y.; Patel, M.; Van Gool, L.; Oswald, M.; Tombari, F. Splat-SLAM: Globally optimized RGB-only SLAM with 3D Gaussians. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 10–17 June 2025.
9. Huang, H.; Li, L.; Cheng, H.; Yeung, S.K. Photo-SLAM: Real-time simultaneous localization and photorealistic mapping for monocular stereo and RGB-D cameras. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024.
10. Chen, S. Gaussian Splatting SLAM (MonoGS). In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024.
11. Hu, Y.S.; Abboud, N.; Ali, M.Q.; Yang, A.S.; Elhajj, I.; Asmar, D.; Chen, Y.; Zelek, J.S. MGSO: Monocular real-time photometric SLAM with efficient 3D Gaussian splatting. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025.
12. Lee, B.; Park, J.; Giang, K.T.; Jo, S.; Song, S. MVS-GS: High-Quality 3D Gaussian Splatting Mapping via Online Multi-View Stereo. *IEEE Access* **2025**, *13*, 1–13. [CrossRef]
13. Lee, B.; Park, J.; Giang, K.T.; Song, S. Online 3D Gaussian Splatting Modeling with Novel View Selection. *arXiv* **2025**, arXiv:2508.14014. [CrossRef]
14. Teed, J.; Deng, J. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 16558–16569.
15. Wen, B.; Trepte, M.; Aribido, J.; Kautz, J.; Gallo, O.; Birchfield, S. FoundationStereo: Zero-shot stereo matching. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 10–17 June 2025.
16. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. [CrossRef]
17. Wang, W.; Zhu, D.; Wang, X.; Hu, Y.; Qiu, Y.; Wang, C.; Hu, Y.; Kapoor, A.; Scherer, S.; TartanAir: A dataset to push the limits of visual SLAM. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 4909–4916.
18. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]

19. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262.
20. Qin, T.; Li, P.; Shen, S. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020.
21. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 611–625. [CrossRef] [PubMed]
22. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
23. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; et al. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Basel, Switzerland, 26–29 October 2011; pp. 127–136.
24. Hornung, A.; Wurm, K.M.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **2013**, *34*, 189–206. [CrossRef]
25. Chang, J.; Chen, Y. Pyramid stereo matching network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5410–5418.
26. Zhang, F.; Prisacariu, V.; Yang, R.; Torr, P.H.S. GA-Net: Guided aggregation net for end-to-end stereo matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 185–194.
27. Lipson, V.; Teed, E.; Deng, J. RAFT-Stereo: Multilevel recurrent field transforms for stereo matching. In Proceedings of the 2021 International Conference on 3D Vision (3DV), London, UK, 1–3 December 2021; pp. 218–227.
28. Xu, J.; Zhang, Z.; Chen, J.; Wang, L. IGEV-Stereo: Iterative geometry encoding volume for stereo matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 21919–21928.
29. Bartolomei, L.; Tosi, F.; Poggi, M.; Mattoccia, S. Stereo Anywhere: Robust zero-shot deep stereo matching even where either stereo or mono fail. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 10–17 June 2025; pp. 1013–1027.
30. Yang, L.; Kang, B.; Huang, Z.; Zhao, Z.; Xu, X.; Feng, J.; Zhao, H. Depth Anything v2. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 21875–21911.
31. Wang, Z.; Liu, S.; Zhu, L.; Chen, H.; Lee, G.H. NICE-SLAM: Neural implicit scalable encoding for SLAM. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12786–12796.
32. Zhu, Y.; Peng, Y.; Wang, Z.; Liu, S.; Lee, G.H. NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 1–5 October 2023.
33. Ma, Y.; Lv, J.; Wei, J. High-Precision Visual SLAM for Dynamic Scenes Using Semantic–Geometric Feature Filtering and NeRF Maps. *Electronics* **2025**, *14*, 3657. [CrossRef]
34. Wei, W.; Wang, J.; Xie, X.; Liu, J.; Su, P. Real-Time Dense Visual SLAM with Neural Factor Representation. *Electronics* **2024**, *13*, 3332. [CrossRef]
35. Keetha, N.; Karhade, J.; Jatavallabhula, K.M.; Yang, G.; Scherer, S.; Ramanan, D.; Luiten, J. SplaTAM: Splat, Track & Map 3D Gaussians for dense RGB-D SLAM. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 21357–21366.
36. Li, M.; Liu, S.; Zhou, H.; Zhu, G.; Cheng, N.; Deng, T.; Wang, H. SGS-SLAM: Semantic Gaussian splatting for neural dense SLAM. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2024; pp. 163–179.
37. Song, S.; Kim, D.; Choi, S. View path planning via online multiview stereo for 3D modeling of large-scale structures. *IEEE Trans. Robot.* **2021**, *38*, 372–390. [CrossRef]
38. Song, S.; Truong, K.G.; Kim, D.; Jo, S. Prior depth-based multi-view stereo network for online 3D model reconstruction. *Pattern Recognit.* **2023**, *136*, 109198. [CrossRef]
39. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
40. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.
41. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]