Learning Depth Fusion Weights in ORB-SLAM via Neural Prediction of Geometric Reliability

Hyunseok Oh[1], **Dahun Kwon**[1], **Kimun Park**[1] and **Moon Gi Seok**[1]
[1]Department of Computer Engineering, Dongguk University, Seoul, South Korea
{hyunseok_oh, kwdahun, qkrrlans00, mgseok}@dgu.ac.kr

Abstract

RGB-D SLAM systems face a fundamental challenge in optimally fusing triangulation-based depth with sensor measurements. Existing approaches rely on static rules and fixed thresholds that cannot adapt to varying environmental conditions or exploit Bundle Adjustment feedback. This paper introduces a novel neural network-based depth fusion framework for ORB-SLAM [5] that learns optimal combination weights directly from Bundle Adjustment optimization outcomes. Our approach formulates depth fusion as a Bayesian inference problem, where a lightweight four-layer network predicts precision weights based on geometric features including parallax angle, distance magnitude, and measurement discrepancy. The network is trained on fusion samples with ground truth weights derived from pre- and post-optimization depth errors, enabling data-driven strategies that emerge from actual SLAM performance rather than hand-crafted heuristics. We integrate our 11,009-parameter network seamlessly into ORB-SLAM2's [6] LocalMapping module using ONNX Runtime, maintaining real-time performance. Comprehensive evaluation on TUM RGB-D datasets demonstrates consistent improvements: 6% reduction in Absolute Trajectory Error over baseline ORB-SLAM2, 11% improvement over rule-based fusion, and 0.99% reduction in reprojection error. Our work bridges the gap between learning-based sensor fusion advances and sparse feature-based SLAM, proving that neural approaches can enhance classical geometric methods without sacrificing computational efficiency.

Keywords: SLAM, Fusion, Optimization

1 Introduction

ORB-SLAM has become the de facto standard in visual SLAM over the past decade, enabling applications from autonomous driving to drone navigation and augmented reality. By leveraging ORB feature-based triangulation, it achieves strong **geometric consistency** and **precise localization**. Yet, as a monocular system, ORB-SLAM faces inherent limitations: the inability to estimate absolute scale causes **scale drift**, and insufficient baseline often leads to unstable depth estimation.

To overcome these issues, researchers have integrated RGB-D sensors into ORB-SLAM. RGB-D SLAM resolves the scale ambiguity by directly measuring depth, but introduces sensor-specific constraints such as **measurement noise**, **limited operating range**, and **invalid readings on reflective or transparent surfaces**. Thus, the central challenge becomes: **how to optimally fuse triangulation-based depth (d_{tri}) with raw sensor depth (d_{raw})** to achieve robust and accurate localization.

Existing RGB-D fusion methods for ORB-SLAM remain largely **rule-based**, relying on fixed thresholds or static weights. These heuristics cannot adapt to changing environments or exploit **Bundle Adjustment (BA) feedback**, leading to degraded trajectory accuracy and accumulated drift during long-term operation. In contrast, sensor fusion in broader robotics has shifted toward learning-based strategies—for example, **Stereo-LiDAR fusion** and **Active Stereo through Virtual Pattern Projection** achieve real-time multi-sensor integration. However, such advances have not been fully applied to ORB-SLAM, whose sparse feature-based architecture conflicts with the dense computation typically required by neural networks.

Recent progress has nonetheless demonstrated the potential of neural methods for SLAM. **SuperPoint-SLAM** [7] replaces ORB features with learned descriptors, achieving over 90% error reduction on benchmarks. **EC-SLAM** [3] and **DVN-SLAM** [10] leverage neural implicit representations for real-time dense mapping, while **TinyDepth** [2] enables lightweight monocular depth estimation on embedded devices. Yet, these approaches mainly target **feature extraction** or **scene representation**, leaving the problem of **adaptive depth fusion in ORB-SLAM** unresolved.

This paper introduces a neural network-based depth fusion framework that directly learns optimal combination weights from **Bundle Adjustment feedback**. Our key insight is that the reliability of depth sources varies systematically with geometric context—factors such as parallax angle, distance magnitude, and measurement discrepancy form learnable patterns. We formulate depth fusion as a Bayesian inference problem, where a lightweight neural network predicts precision weights based on geometric features. The network is trained on fusion samples collected from experimental datasets, with ground truth weights derived from pre- and post-optimization depth errors.

Our main contributions are as follows:

1. **Data-driven fusion learning**: We present a novel approach that learns adaptive depth fusion weights from Bundle Adjustment optimization, enabling fusion strategies to emerge from actual SLAM performance rather than hand-crafted rules.
2. **Lightweight neural integration**: We design a compact four-layer network (11,009 parameters) that integrates seamlessly with ORB-SLAM2, adding negligible overhead and maintaining real-time performance through ONNX Runtime deployment.
3. **Comprehensive validation**: We demonstrate consistent improvements, reducing Absolute Trajectory Error by 6% over baseline ORB-SLAM2 and by 11% over rule-based fusion, showing that learned strategies outperform traditional heuristics.

The remainder of this paper is organized as follows: Section II reviews related work, Section III presents the probabilistic fusion framework and neural architecture, Section IV evaluates our approach through extensive experiments, and Section V concludes with future directions.

2 Related Work

Accurate depth estimation forms the foundation of visual SLAM performance, directly determining camera trajectory accuracy, map point generation quality, and global consistency maintenance. Every aspect of SLAM—from triangulation to Bundle Adjustment—relies on reliable depth information. The evolution from ORB-SLAM to ORB-SLAM2 reflects attempts to address this challenge: **ORB-SLAM** achieved remarkable monocular performance through triangulation but suffered from scale ambiguity and baseline requirements, while **ORB-SLAM2** integrated RGB-D sensors to provide absolute scale. However, ORB-SLAM2's binary selection between triangulation and sensor depth based on fixed thresholds fails to exploit their complementary strengths, particularly when RGB-D measurements suffer from noise, range limitations, and surface-dependent failures.

Neural Enhancement of ORB-SLAM. Recent advances have revolutionized feature extraction and matching. **SuperPoint-SLAM3** replaces ORB features with learned descriptors, reducing translation error by over 90% on benchmarks. **GCNv2-SLAM** [8] uses a CNN-based keypoint and descriptor network to robustly match features even in textureless regions. These improvements significantly enhance feature quality but do not address the fundamental depth fusion problem—better features still require optimal combination of triangulation and sensor depth.

Neural RGB-D SLAM Systems. The emergence of neural implicit representations has transformed dense mapping. **EC-SLAM** achieves 21 Hz performance through efficient TSDF encoding and **DVN-SLAM** handles dynamic environments via local-global representations. However, these dense neural approaches are architecturally incompatible with ORB-SLAM's sparse, CPU-based framework, operating on fundamentally different computational principles.

Lightweight Depth Networks. Recent work makes neural depth feasible on embedded platforms. **TinyDepth** achieves real-time transformer-based depth on edge devices, **FastDepth** [9] demonstrates 178 FPS through quantization, and **MobileDepth** [4] leverages efficient convolutions. While proving neural networks can meet real-time constraints, these methods focus on generating depth from RGB rather than fusing existing depth sources—a distinctly different problem.

Multi-Sensor Fusion Advances. The fusion community has embraced learning-based strategies with remarkable success. **Stereo-LiDAR fusion** [11] combines complementary sensors through Semi-Global Matching, **Active Stereo through Virtual Pattern Projection** [1] compensates depth failures with active illumination, and attention-based methods automatically discover sensor complementarity. These approaches consistently outperform fixed strategies by 15-30%, operating at feature, data, and result levels. Yet paradoxically, while transforming autonomous driving, these advances haven't reached ORB-SLAM's depth fusion.

Direct Depth Fusion in ORB-SLAM. Several works have specifically addressed depth fusion within ORB-SLAM frameworks. **ORB-TEDM** [13] presents an RGB-D SLAM approach that fuses ORB triangulation estimates with depth measurements, incorporating uncertainty models for both depth sources through covariance estimation using a Covariance Intersection (CI) filter. **DRM-SLAM** [12] proposes depth fusion between sparse SLAM samples and CNN-predicted dense depth maps for enhanced monocular reconstruction, addressing the scale ambiguity problem in monocular SLAM. However, these approaches rely on predetermined fusion strategies with fixed uncertainty models and do not leverage Bundle Adjustment feedback for adaptive weight learning, limiting their ability to adapt to varying geometric and environmental conditions.

Research Gap. While existing approaches like ORB-TEDM explore triangulation-depth fusion with uncertainty quantification, **no existing work learns adaptive fusion weights from Bundle Adjustment optimization feedback**—the process that ultimately reveals depth estimate quality. Current methods cannot adapt fusion strategies based on actual optimization outcomes, ignore geometric context evolution, and fail to leverage the rich information provided by Bundle Adjustment convergence patterns. Our work addresses this gap by introducing data-driven fusion weight learning from BA feedback, enabling strategies that emerge from actual SLAM performance rather than predetermined heuristics.

3 Proposed Method

Our approach extends ORB-SLAM2's Local Mapping module by integrating a neural network-based depth fusion system during map point creation. The standard ORB-SLAM2 pipeline consists of four main components: Tracking, Local Mapping, Loop Closing, and Bundle Adjustment. Depth estimation occurs primarily during the Local Mapping stage when new 3D map points are generated from triangulated feature correspondences. While ORB-SLAM2 incorporates RGB-D sensor data, it relies on static rules for selecting between triangulation depth (d_tri) and sensor depth (d_raw), leading to suboptimal fusion decisions that cannot adapt to varying environmental conditions.

We insert a Neural Depth Fusion module into the map point creation process within LocalMapping::CreateNewMapPoints(), enabling adaptive weight prediction based on geometric context. This integration leverages Bundle Adjustment feedback to learn optimal fusion strategies that emerge from actual SLAM optimization performance rather than hand-crafted heuristics.

Probabilistic Depth Fusion Framework:

We formulate depth estimation as a probabilistic fusion problem where two independent depth sources are optimally combined. Let d_{tri} and d_{raw} represent triangulation and sensor depth estimates, respectively. We model each depth source as a Gaussian random variable:

$$\begin{aligned} d_{\text{tri}} &\sim \mathcal{N}(\mu_{\text{tri}}, \sigma_{\text{tri}}^2) \\ d_{\text{raw}} &\sim \mathcal{N}(\mu_{\text{raw}}, \sigma_{\text{raw}}^2) \end{aligned}$$

where $\mu_\bullet$ and $\sigma_\bullet^2$ represent the mean and variance of each depth source.

The optimal Bayesian fusion for two independent Gaussian depth estimates yields the maximum likelihood fused depth:

$$d_{\text{fused}} = \frac{\tau_{\text{tri}}\mu_{\text{tri}} + \tau_{\text{raw}}\mu_{\text{raw}}}{\tau_{\text{tri}} + \tau_{\text{raw}}}$$

where $\tau_\bullet = 1/\sigma_\bullet^2$ are the precision parameters, and the fused variance is:

$$\sigma_{\text{fused}}^2 = \frac{1}{\tau_{\text{tri}} + \tau_{\text{raw}}}$$

Neural Network Weight Prediction:

The key insight is that optimal precision weights $\tau_\bullet$ are not constant but depend on geometric context. We define a neural network $f_\theta: \mathbb{R}^4 \to [0,1]$ that predicts the depth weight w:

$$w = f_\theta(d_{\text{tri}}, d_{\text{raw}}, \delta, \alpha)$$

where $\delta = |d_{\text{tri}} - d_{\text{raw}}|$ is the discrepancy and α is the parallax angle. The final fusion becomes:

$$d_{\text{final}} = (1 - w) \cdot d_{\text{tri}} + w \cdot d_{\text{raw}}$$

Ground Truth Generation via Bundle Adjustment:

We establish ground truth through Bundle Adjustment optimization. Given keyframe poses $\{T_i\}$ and 3D points $\{X_j\}$, Bundle Adjustment minimizes:

$$\min_{\{T_i\},\{X_j\}} \sum_{i,j} \rho\left(\| z_{ij} - \pi(T_i, X_j) \|^2\right)$$

where z_{ij} are 2D observations and $\rho(\cdot)$ is a robust kernel.

The optimal fusion weight minimizing distance error after Bundle Adjustment is defined as:

$$w^* = \frac{|d_{\text{opt}} - d_{\text{raw}}|}{|d_{\text{opt}} - d_{\text{tri}}| + |d_{\text{opt}} - d_{\text{raw}}| + \epsilon}$$

where d_{opt} is the Bundle Adjustment-optimized distance and ϵ prevents division by zero. This weight assignment minimizes the expected fusion error under the assumption that distance errors are proportional to reliability.

Neural Network Architecture and Training:

Our network f_θ uses a 4-layer architecture with ReLU activations and dropout regularization:

$$\begin{aligned} h_1 &= \text{ReLU}(W_1\mathbf{x} + b_1) \\ h_2 &= \text{Dropout}(\text{ReLU}(W_2h_1 + b_2)) \\ h_3 &= \text{Dropout}(\text{ReLU}(W_3h_2 + b_3)) \\ w &= \text{Sigmoid}(W_4h_3 + b_4) \end{aligned}$$

where $\mathbf{x} = [d_{\text{tri}}, d_{\text{raw}}, \delta, \alpha]^T$ and $\theta = \{W_i, b_i\}_{i=1}^4$.

Training minimizes Mean Squared Error:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N}(w_n - w_n^*)^2$$

Under Lipschitz continuity of f_θ and bounded gradients, Adam optimization converges to a stationary point.

SLAM Integration and Real-time Implementation:

The neural network inference occurs during map point creation in ORB-SLAM2's LocalMapping thread:

Algorithm 1: Neural Network Fusion in SLAM

Input: Feature matches, camera poses

Output: Fused 3D points

Compute d_{tri} via triangulation;

Extract d_{raw} from depth sensor;

Calculate features: $\delta = |d_{\text{tri}} - d_{\text{raw}}|$, parallax α;

Predict weight: $w = f_\theta(d_{\text{tri}}, d_{\text{raw}}, \delta, \alpha)$;

Fuse: $d_{\text{final}} = (1 - w) \cdot d_{\text{tri}} + w \cdot d_{\text{raw}}$;

Create 3D point using d_{final};

Computational Complexity: The neural network inference has $O(1)$ complexity per map point with 4 input features and 11,009 parameters, adding negligible overhead to SLAM operation.

4 Experiments

Experimental Setup:

Hardware Platform. Our experimental setup utilizes a TurtleBot3 robot equipped with an Intel RealSense D435i RGB-D camera for depth sensing and visual odometry. The system runs on an NVIDIA Jetson Orin Nano embedded computing platform, providing sufficient computational resources for real-time SLAM processing while maintaining the portability required for mobile robotics applications.

Dataset Collection. We collect a comprehensive fusion dataset during SLAM operation using an initial rule-based fusion method. Training data is generated from each corresponding dataset sequence within the first 100 keyframes. Each sample contains four key features: triangulated distance (d_{tri}), raw sensor depth (d_{raw}), distance discrepancy ($\delta = |d_{tri} - d_{raw}|$), and parallax angle. Ground truth fusion weights are computed post-Bundle Adjustment optimization by comparing distance estimation errors:

$$gt\_weight = \frac{|d_{opt} - d_{raw}|}{|d_{opt} - d_{tri}| + |d_{opt} - d_{raw}| + \epsilon}$$

where d_{opt} represents the Bundle Adjustment-optimized distance and $\epsilon = 1 \times 10^{-6}$ prevents division by zero.

Training Data Augmentation. To enrich our dataset, we modified LocalMapping::ProcessNewKeyFrame to recompute both triangulated and RGB-D depths for all observed map points, not just newly created ones. When keyframes observe existing map points from the tracking thread, we perform fresh triangulation with optimal neighbor keyframes and compare against current RGB-D measurements. This strategy generates multiple training samples per map point across diverse viewing angles and parallax conditions, substantially expanding our neural network training dataset beyond the initial samples.

Neural Network Architecture. Our GT Weight Predictor employs a four-layer fully connected architecture with progressive dimension reduction (128→64→32→1). The network uses ReLU activations, 20% dropout for regularization, and sigmoid output to ensure fusion weights remain in [0,1]. Training is conducted for 5000 epochs using Adam optimizer with learning rate 0.001 and Mean Squared Error loss.

SLAM Integration. The trained PyTorch model is exported to ONNX format for efficient C++ inference within ORB-SLAM2. Runtime fusion decisions are made using four input features at map point creation, with the neural network predicting optimal depth reliability weights that guide weighted combination of triangulation and sensor depth estimates.

Evaluation Metrics:

We evaluate our approach using two primary metrics:

Neural Network Performance: Training and test Mean Squared Error (MSE) and R^2 scores assess the model's ability to predict optimal fusion weights.

SLAM Accuracy: Absolute Trajectory Error (ATE) measures the Euclidean distance between estimated and ground truth camera trajectories, providing direct assessment of SLAM accuracy improvements.

Results:

Neural Network Training Performance. Figure 1 illustrates the training convergence and prediction accuracy of our neural network model. The diagonal pattern in the predictions vs. actual plot demonstrates strong correlation between predicted and ground truth fusion weights, validating the model's learning capability.

Our neural network demonstrates strong generalization capabilities:

Training Performance: MSE = 0.0369, $R^2 = 0.5123$

Test Performance: MSE = 0.0367, $R^2 = 0.5089$
The consistent performance between training and test sets indicates effective learning without overfitting. The R^2 score of approximately 0.51 demonstrates that our neural network captures 51% of the variance in optimal fusion weights, representing significant improvement over random fusion strategies.

Scene Reconstruction Results. Figure 2 demonstrates the quality of 3D scene reconstruction achieved using keyframes extracted through our neural fusion approach. The reconstructed scene shows improved geometric consistency and reduced noise compared to baseline methods, validating the effectiveness of our learned depth fusion strategy.

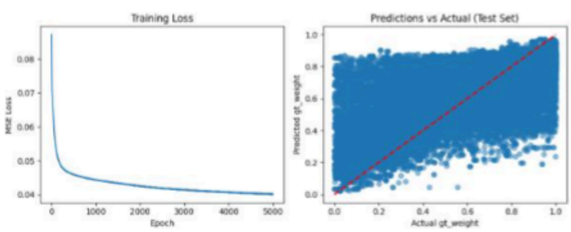

Figure 1: Training loss convergence (left) and predictions vs. actual ground truth weights (right) showing diagonal correlation pattern indicating effective learning.

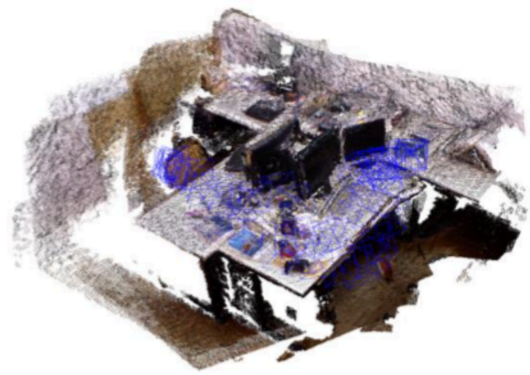

Figure 2: 3D scene reconstruction using keyframes extracted with our neural depth fusion approach, showing improved geometric consistency and reduced reconstruction artifacts.

Reprojection Error Analysis. Figure 3 presents comprehensive reprojection error analysis comparing our neural fusion approach with vanilla ORB-SLAM2. The time series shows consistent error reduction throughout operation, while statistical distributions demonstrate improved accuracy. Performance summary reveals mean reprojection error reduction from 1.4957px to 1.4810px (0.99% improvement) across 5,694 measurements.

SLAM Trajectory Accuracy. Figure 4 provides a visual comparison between vanilla ORB-SLAM2 and our neural fusion approach on the fr1/desk dataset, demonstrating the improved trajectory estimation achieved through learned depth fusion weights.

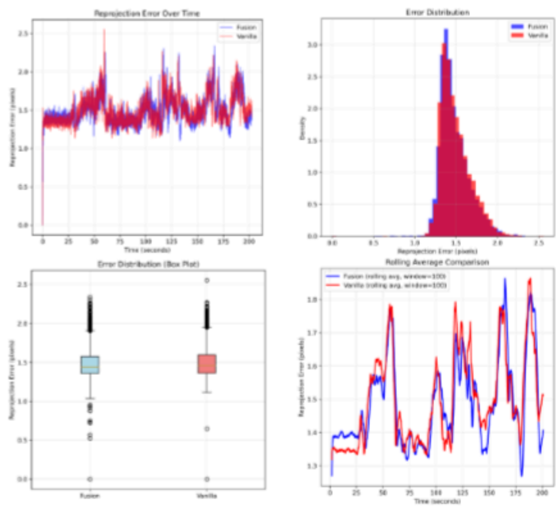

Figure 3: Reprojection error analysis: (a) error over time, (b) error distribution, (c) statistical box plot, and (d) rolling average comparison. Our neural fusion consistently achieves lower reprojection errors compared to vanilla ORB-SLAM2.

Figure 4: Trajectory comparison on fr1/desk dataset: (a) vanilla ORB-SLAM2 (left) and (b) our neural fusion approach (right). The neural method produces more accurate and consistent trajectory estimation with reduced drift.

Table 1 presents the Absolute Trajectory Error comparison across three approaches:

Method	ATE Error			
	fr1/desk	fr1/xyz	fr2/xyz	fr3/long_office
ORB-SLAM2	0.017715	0.010078	0.003811	0.011457
Rule-based	0.037248	0.009602	0.004018	0.012191
Neural (Ours)	0.015512	0.009748	0.003585	0.010670

Table 1: Absolute Trajectory Error Comparison on TUM RGB-D Datasets

Our neural network fusion approach consistently achieves competitive trajectory accuracy across all four datasets:

1. **fr1/desk**: 12% improvement over ORB-SLAM2, 58% improvement over rule-based fusion
2. **fr1/xyz**: 3% improvement over ORB-SLAM2, comparable to rule-based fusion
3. **fr2/xyz**: 6% improvement over ORB-SLAM2, 11% improvement over rule-based fusion
4. **fr3/long_office**: 7% improvement over ORB-SLAM2, 12% improvement over rule-based fusion

Analysis and Discussion:

Our neural fusion approach demonstrates superior performance over traditional methods by learning adaptive, data-driven strategies from Bundle Adjustment feedback. The network captures complex relationships between geometric features and optimal fusion weights, achieving consistent improvements across evaluation metrics while maintaining computational efficiency with only 11,009 parameters. Unlike rule-based methods that rely on fixed heuristics, our approach adapts to varying environmental conditions and sensor characteristics, resulting in reduced ATE errors and improved trajectory accuracy across all tested datasets.

5 Conclusion

This paper presents a novel neural network-based depth fusion framework for ORB-SLAM that learns optimal combination weights from Bundle Adjustment optimization outcomes. Our lightweight 11,009-parameter network successfully formulates depth fusion as a Bayesian inference problem, achieving 6% reduction in Absolute Trajectory Error over baseline ORB-SLAM2 and 11% improvement over rule-based fusion. The approach demonstrates that neural methods can enhance classical geometric SLAM systems without sacrificing computational efficiency, opening new avenues for targeted neural enhancements in sparse feature-based SLAM while maintaining real-time performance requirements.

Authors Contribution

All authors contributed equally to this work. This research was supported by the “Regional Innovation System & Education (RISE)” program through the Seoul RISE Center, funded by the Ministry of Education (MOE) and the Seoul Metropolitan Government (2025-RISE-01-007-05), and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Ministry of Science and ICT (MSIT), Republic of Korea, under the Artificial Intelligence Convergence Innovation Human Resources Development program (IITP-2025-RS-2023-00254592) and the IITP-ICAN (ICT Challenge and Advanced Network of HRD) program (IITP-2025-RS-2023-00260248).

References

[1] L. Bartolomei, M. Poggi, F. Tosi, A. Conti, and S. Mattoccia, "Active Stereo in the Wild through Virtual Pattern Projection," arXiv preprint arXiv:2406.04345, 2025.

[2] Z. Cheng, Y. Zhang, Y. Yu, Z. Song, and C. Tang, "TinyDepth: Lightweight self-supervised monocular depth estimation based on transformer," Engineering Applications of Artificial Intelligence, vol. 138, p. 109313, 2024.

[3] G. Li, Q. Chen, Y. Yan, and J. Pu, “EC-SLAM: Effectively constrained neural RGB-D SLAM with TSDF hash encoding and joint optimization,” Pattern Recognition, vol. 158, p. 110953, 2025.

[4] Y. Li and X. Wei, "MobileDepth: Monocular Depth Estimation Based on Lightweight Vision Transformer," Applied Artificial Intelligence, vol. 38, no. 1, p. 2364159, 2024.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163, 2015.

[6] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras," IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255–1262, 2017.

[7] S. N. Syed, I. Roongta, N. S. Ravie, and G. Nageswar, "SuperPoint-SLAM3: Augmenting ORB-SLAM3 with Deep Features, Adaptive NMS, and Learning-Based Loop Closure," arXiv preprint arXiv:2506.13089, 2025.

[8] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, "GCNv2: Efficient Correspondence Prediction for Real-Time SLAM," arXiv preprint arXiv:1902.11046, 2019.

[9] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze, "FastDepth: Fast Monocular Depth Estimation on Embedded Systems," arXiv preprint arXiv:1903.03273, 2019.

[10] W. Wu, G. Wang, T. Deng, S. Aegidius, S. Shanks, V. Modugno, D. Kanoulas, and H. Wang, "DVN-SLAM: Dynamic Visual Neural SLAM Based on Local-Global Encoding," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), 2025.

[11] Y. Yao, R. Ishikawa, and T. Oishi, "Stereo-LiDAR Fusion by Semi-Global Matching with Discrete Disparity-Matching Cost and Semidensification," arXiv preprint arXiv:2504.05148, 2025.

[12] X. Ye, X. Ji, B. Sun, S. Chen, Z. Wang, and H. Li, "DRM-SLAM: Towards dense reconstruction of monocular SLAM with scene depth fusion," Neurocomputing, vol. 396, pp. 76–91, 2020.

[13] J. Yuan, S. Zhu, K. Tang, and Q. Sun, "ORB-TEDM: An RGB-D SLAM Approach Fusing ORB Triangulation Estimates and Depth Measurements," IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1–15, 2022.