# RGB-Only 3D Scene Reconstruction via Uncertainty-Aware 3D Gaussian SLAM

Project ID: PRJ-NTPUCSIE-113-009

Project Duration: From September, 2024 to May, 2025

**Chih-Yao Chang**
*Computer Science Department*
*National Taipei University*
*New Taipei, Taiwan*

**Tzu-Chieh Chen**
*Computer Science Department*
*National Taipei University*
*New Taipei, Taiwan*

**Yan-Ting Li**
*Computer Science Department*
*National Taipei University*
*New Taipei, Taiwan*

**Jia-You Xiao**
*Computer Science Department*
*National Taipei University*
*New Taipei, Taiwan*

***Abstract*—3D reconstruction is a fundamental task in computer graphics, with broad applications in virtual reality, gaming, and cultural heritage preservation. However, traditional methods such as LiDAR scanning are often hindered by high costs and limited operational range, restricting their accessibility and scalability. To overcome these challenges, we present a novel, effective approach that leverages RGB video input for accurate 3D reconstruction. Our method integrates 3D Gaussian representations with Simultaneous Localization and Mapping (SLAM) techniques to generate high-fidelity models from conventional video. This framework offers a practical and accessible solution for high-quality 3D reconstruction.**

## I. Introduction

Traditional dense Simultaneous Localization and Mapping (SLAM) approaches often rely on explicit handcrafted representations such as points, surfels, or signed distance fields. While these methods have reached production-level maturity, they struggle with capturing unobserved viewpoints and require high-frame-rate, geometrically rich input. More recently, neural implicit volumetric representations—such as those based on radiance fields—have emerged, offering impressive visual fidelity through differentiable rendering. However, they are typically computationally intensive, hard to edit, and lack explicit geometry modeling.

Visual Simultaneous Localization and Mapping (SLAM) is a critical capability for autonomous systems, enabling real-time pose estimation and 3D environment mapping. A key factor influencing SLAM performance is the choice of map representation, which significantly affects system efficiency, accuracy, and downstream applications.

To address these challenges, SplaTAM [1] propose a SLAM framework built on explicit volumetric 3D Gaussians, which it use to Splat, Track, and Map. This representation combines the advantages of fast rasterization-based rendering (up to 400 FPS), spatially explicit mapping, and efficient photometric optimization.

Despite its strengths, SplaTAM relies on depth input, which in its original implementation is obtained using a LiDAR sensor. However, in our experiments, we observed that the LiDAR used by SplaTAM can only capture reliable depth information within a short range of approximately 1 to 1.5 meters. This limitation makes SplaTAM unsuitable for reconstructing larger-scale scenes. To address this, we replace the LiDAR depth with predicted depth maps derived from RGB inputs. Most importantly, we propose a novel method to ensure that the use of predicted depth does not compromise reconstruction accuracy, enabling our system to maintain high-fidelity scene representations.

## II. Realted work

### A. Traditional dense SLAM methods

Traditional dense SLAM methods have investigated a range of explicit representations for modeling 3D scenes. These include 2.5D images that capture partial 3D geometry through depth or height maps [9], Gaussian mixture models [11], (truncated) signed distance functions (SDFs) [10], and circular surfels [12]. Circular surfels—colored, disk-shaped surface elements—have proven particularly effective for real-time optimization from RGB-D inputs. While surfel-based approaches support efficient scene reconstruction, they are inherently discontinuous and thus require careful regularization to mitigate artifacts such as holes in the reconstructed geometry [12].

Recent advances in differentiable rendering, such as differentiable rasterizers [13], facilitate gradient flow through depth discontinuities and improve optimization. However, traditional SLAM systems often rely on non-differentiable visibility functions, limiting their compatibility with gradient-based optimization techniques. In contrast, our work adopts a volumetric scene representation based on 3D Gaussians, rather than surface-based primitives, to enable smooth, continuous optimization for efficient and accurate SLAM.

### B. Pretrained neural network representations

Recent advancements have integrated pretrained neural network representations with traditional SLAM techniques, primarily focusing on predicting depth from RGB images to enhance mapping and localization. Early approaches directly incorporate neural network-predicted depth maps into SLAM pipelines [14], providing a straightforward way to leverage learned depth information. More sophisticated methods employ variational autoencoders (VAEs) to decode compact, optimizable latent codes into depth maps [15], enabling efficient representation and optimization. Other techniques

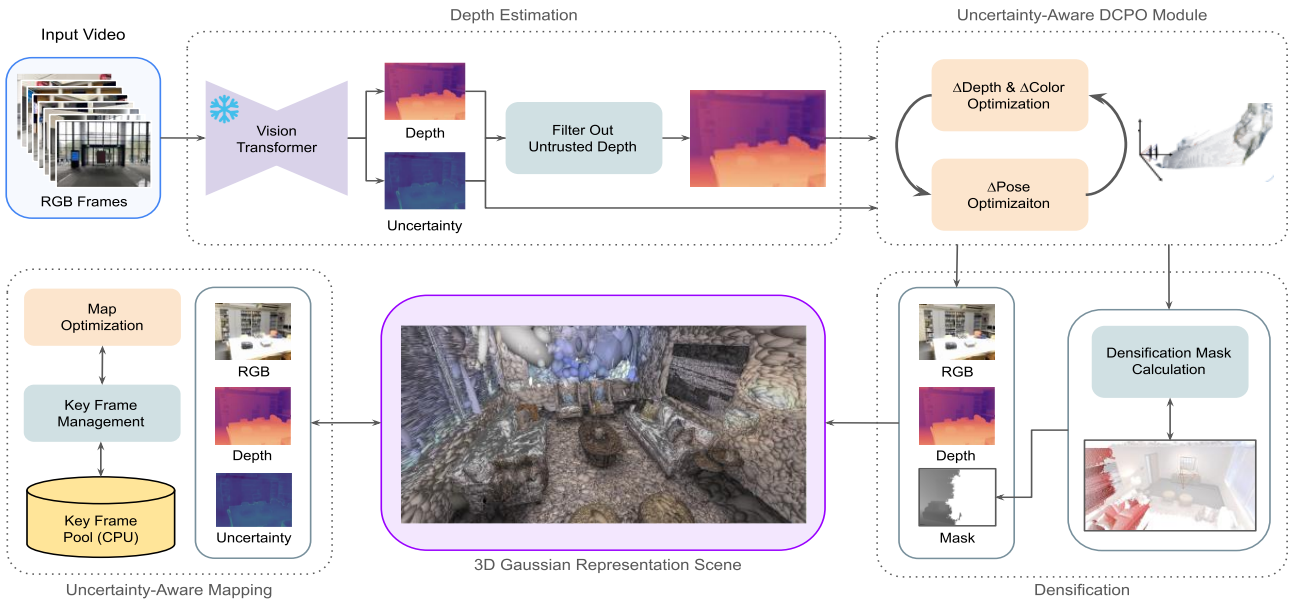


Fig.1. System Architecture Diagram

simultaneously learn to predict depth cost volumes and perform tracking [16], combining depth estimation with camera motion estimation in a unified framework.

### C. 3D Gaussian Splatting

Recently, 3D Gaussians have emerged as a powerful representation for 3D scene modeling, enabling high-speed, differentiable rendering through splatting techniques [17]. This representation has been extended to dynamic scene modeling by incorporating dense six-degree-of-freedom(6-DOF) motion, facilitating applications across both static and dynamic environments [18]. Despite their effectiveness, these methods typically assume access to accurate 6-DOF camera poses for each input frame in order to optimize the scene representation [18]. Such reliance on precomputed poses limits their applicability in real-world SLAM scenarios, where camera poses may be unknown or corrupted by noise. In this work, we address this limitation by jointly estimating camera poses and fitting the underlying Gaussian representation, thereby eliminating the need for externally provided pose information.

## III. Method

Our method (shown in Fig. 1) is broadly inspired by SplaTAM, but differs in a key aspect: unlike SplaTAM, which relies on RGB-D input, our approach reconstructs 3D scenes using RGB images alone. This requires addressing the absence of explicit depth information. To this end, we first predict per-frame depth maps from RGB inputs using a monocular depth estimator. These predicted depths are then refined and aligned with the corresponding RGB images and camera poses through our proposed Depth-Color-Pose Optimization (DCPO) module. This alignment step ensures geometric consistency and improves reconstruction accuracy during the SLAM process.

### A. *Depth Estimation*

Traditional SLAM acquiring depth using LiDAR are expensive or limited by their effective range. In our experiments with the iPhone's LiDAR sensor, we observed that it could only reliably capture depth within approximately 1 to 1.5 meters (shown in Fig. 2 and Fig. 3).

Therefore, we adopted a state-of-the-art (SOTA) monocular depth prediction model. This model generates a dense depth map for each RGB image, along with a per-pixel uncertainty map that quantifies confidence in each prediction. By filtering out high-uncertainty pixels, we improve the reliability of the depth map, which in turn enhances the performance of subsequent processing stages such as pose estimation and scene reconstruction.

Fig.2. From left to right:
Real-world scene / LiDAR depth map / Predicted depth map.

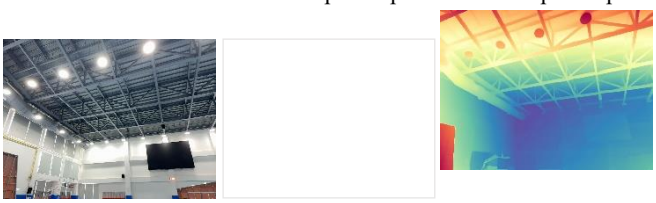

Fig.3. From left to right:
Real-world scene / LiDAR depth map (empty) / Predicted

### B. *Uncertainty-Aware DCPO module*

All three components are optimized within our Uncertainty-Aware Depth-Color-Pose Optimization (DCPO) module using an iterative optimization strategy. Instead of jointly optimizing all parameters at once, which often leads to ambiguity and numerical instability, we alternate between optimizing each component while keeping the others fixed.

By optimizing Depth, RGB, and Pose in a cyclical and decoupled manner, we effectively disentangle these variables, avoid local minima, and achieve more accurate and stable convergence. This not only improves the quality of the final reconstruction, but also leads to faster and more reliable optimization in practice.

**1) Uncertainty-Aware Pose Optimization(Tracking).**

Thanks to 3D Gaussian Splatting, the photometric loss is differentiable with respect to the camera pose. We aim to refine the camera pose by minimize the image and depth reconstruction error of the RGB and Depth frame with respect to camera pose parameters for t + 1, but only evaluate errors over pixels within the visible mask. However, since the predicted depth often contains noise that is unreliable, we incorporate the depth uncertainty map to guide camera pose tracking. Specifically, we down-weight with high uncertainty during the loss calculation. This ensures that the pose refinement process is not biased by unreliable depth values, improving robustness in challenging regions.

**2) Uncertainty-Aware Depth Optimization.**

As the depth is predicted, aligning it with the scene is crucial to preserve structural accuracy in the reconstruction. We optimize a global scale and translation offset for each depth map, aligning it with the 3D scene reconstructed so far (shown in Fig. 4). We also incorporate the depth uncertainty map to down-weight unreliable pixels during this optimization. This ensures that the refinement process focuses on trustworthy regions of the depth map. Together, these steps allow us to produce consistent global depth estimates and better tracking accuracy. In addition to refining the camera pose, we also optimize the predicted depth map using a scale and translation transformation. This is crucial because monocular depth estimation is an ill-posed problem—it lacks absolute scale information. As a result, predicted depth maps often have correct relative structure but inconsistent geometry across frames and hinder reliable tracking and reconstruction.

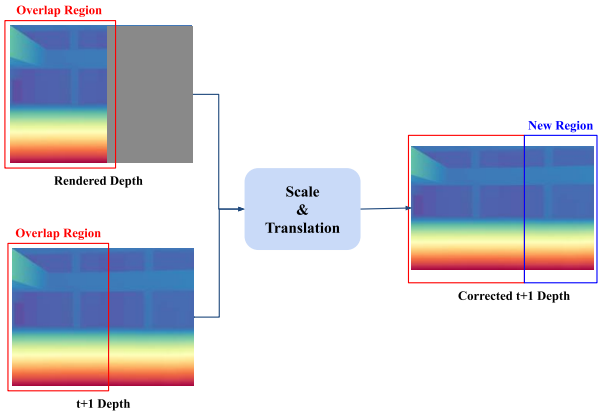


Fig.4. Uncertainty-Aware Depth Optimization

**3) RGB Optimization**

Finally, we also apply a linear color transformation to the RGB frames to account for variations in exposure and white balance. Even within the same scene, differences in lighting conditions—such as camera auto-exposure adjustments or slight changes in viewpoint—can lead to noticeable color shifts between frames. These inconsistencies can degrade the accuracy of photometric tracking and cause discontinuous or distorted reconstructions. By learning a simple linear transformation (scale and bias) per frame, we align the appearance of incoming images with the existing 3D scene (shown in Fig. 5), improving both pose estimation stability and the visual coherence of the final reconstruction.

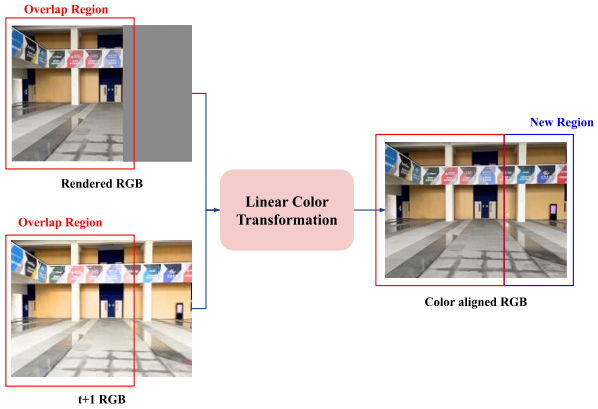


Fig.5. RGB Optimization

## C. *3D Gaussian Densification*

Gaussian Densification (shown in Fig. 6) aims to inject new 3D Gaussians into the scene map as each new frame is processed. After pose tracking, we obtain an accurate camera pose for the current frame, and the corrected depth map provides reliable 3D geometry. This allows us to estimate where new Gaussians should be placed in the world.

However, we want to avoid redundant or unnecessary densification. If the existing Gaussians already represent the scene geometry well, adding more would be inefficient and potentially harmful. To address this, we compute a densification mask—a per-pixel decision map that indicates where new Gaussians are needed.

This mask highlights two main cases:

1) Underdense areas where the map does not yet have sufficient coverage.
2) New geometry in front of existing geometry.

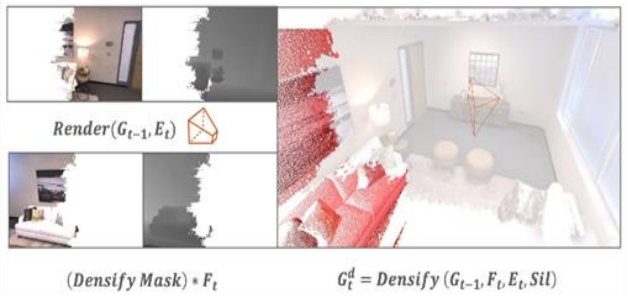


Fig.6. 3D Gaussian Densification

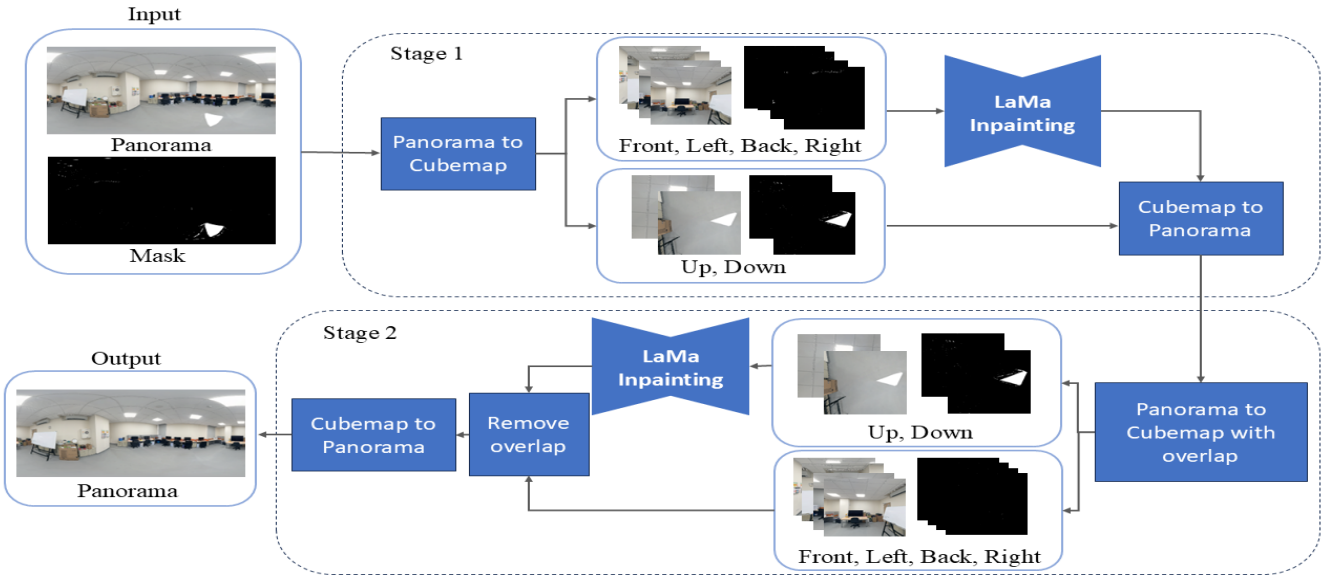


Fig.7. Two-Stage Cubemap-based LaMa Inpainting: Restore horizontal faces first for stability, then use overlap cues to complete top and bottom. By providing overlap information, we enable the top and bottom faces to gain more contextual insight for inpainting. This guides lower-confidence areas using higher-confidence regions.

### *D. Uncertainty-Aware Gaussian Mapping*

This step refines and consolidates the 3D Gaussian Map after tracking. While tracking provides a rough alignment of new observations, it is often noisy and incomplete—especially due to uncertainty in monocular depth predictions. Therefore, we perform a dedicated mapping phase to optimize the position, appearance, and opacity of Gaussians, while keeping camera poses fixed. To address depth unreliability, we use a per-pixel uncertainty map during optimization. Pixels with high uncertainty are down-weighted in the loss, reducing the impact of noisy geometry. This results in a cleaner and more stable reconstruction.

## IV. RESULT

In this section, we first present the reconstruction results of our system on both benchmark and real-world datasets. To further demonstrate the practical utility of our reconstructions, we render 360-degree panoramas with our scenes. Our method reconstructs scenes using only a short RGB video as input, enabling re-located rendering of panoramas from arbitrary positions within the scene. Compared to traditional method, which typically requires expensive hardware or time-consuming image stitching method, this significantly reduces both time and labor while maintaining high visual fidelity.

### *A. Our reconstruction results*

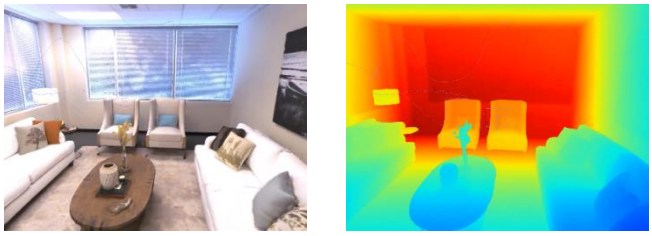

Fig.8. Reconstruction result on benchmark dataset (RGB/Depth)

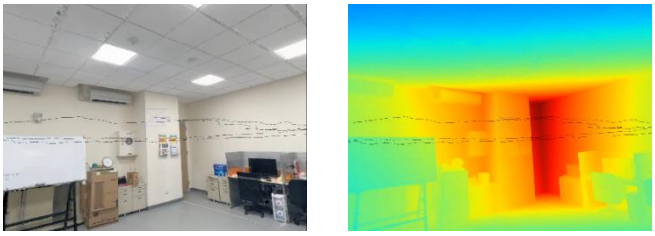

Fig.9. Reconstruction result on real-world dataset (RGB/Depth)

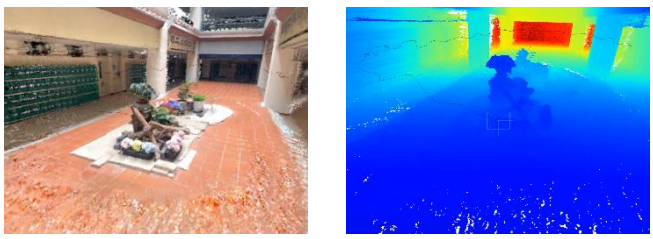

Fig.10. Reconstruction result on real-world dataset (RGB/Depth)

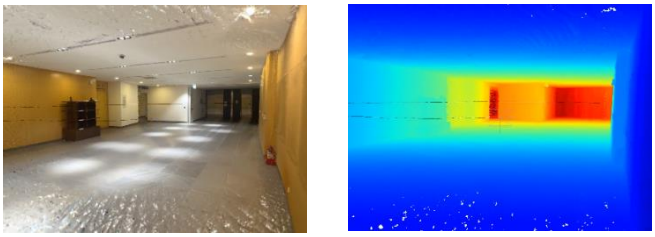

Fig.11. Reconstruction result on real-world dataset (RGB/Depth)


Fig.12. Comparison of 360° panoramas: Comparison shows that the panorama rendered from our scene maintains more accurate geometric structure than the stitching result.

### *B. 360° panoramas results*

Traditional 360° panoramas are generated via image stitching, which requires fixed camera positions and is prone to artifacts from camera motion. In contrast (shown in Fig. 12), our method enables flexible, efficient panorama rendering from arbitrary viewpoints using the reconstructed scene.

During relocation, new viewpoints may reveal hidden areas, requiring inpainting to complete 360° panoramas. We propose a **Two-Stage Cubemap-based LaMa Inpainting** method (shown in Fig. 7) to fill missing regions and enhance panorama quality. Considering the limited effectiveness of current 360° panorama inpainting methods, our approach employs the stable 2D image restoration model, LaMa. Converting the Panorama to a Cubemap representation primarily preserves original perspective geometry, avoiding severe distortions in areas like the zenith and nadir, thus improving LaMa's structural understanding and inpainting accuracy.

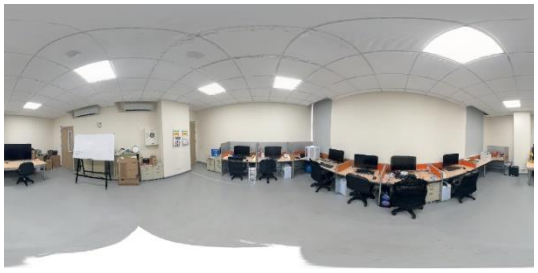

Fig.13. Before inpainting

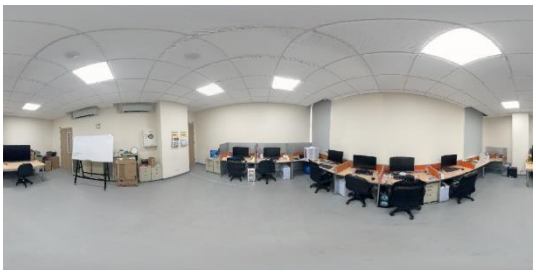

Fig.14. After inpainting

Fig.15. Inpainting result

Fig.16. Inpainting result

We also use 360° panoramas as environmental light in Unity to demonstrate our inpainting result.

Fig.17. Lighting in Unity with our 360° panorama

Fig.18. Lighting in Unity with our 360° panorama

Fig.19. Lighting in Unity with our 360° panorama

## ACKNOWLEDGMENT

We would like to express our sincere gratitude to our professor, for his invaluable guidance and support throughout this project. His contributions can be summarized as follows:

1. Literature Review and Discussion

During the course of the project, professor carefully selected and provided us with a wide range of relevant research papers and literature. Through reading and presenting these materials, we were able to acquire new knowledge and develop innovative ideas.

2. Method Design and Technical Guidance

Whenever we encountered difficulties in designing our method, professor provided insightful discussions and helped us analyze potential causes and directions for improvement, which greatly contributed to the progress of our system.

3. Unity Instruction and Support

As Unity was an essential tool for verifying the correctness of our algorithmic implementation, we often faced technical challenges due to its complexity. Professor patiently guided us through Unity's functionalities and resolved various issues during development.

4. Report Writing and Result Presentation

Professor also instructed us in academic writing, including how to structure the report and effectively present our findings. He provided thorough reviews and constructive feedback to help refine our final documentation, making it clearer and more readable.

We are deeply thankful for his dedication and encouragement throughout every stage of the project.

## REFERENCES

[1] Keetha, N., Karhade, J., Jatavallabhula, K. M., Yang, G., Scherer, S., Ramanan, D., & Luiten, J. (2024). Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 21357-21366).

[2] Piccinelli, L., Sakaridis, C., Yang, Y. H., Segu, M., Li, S., Abbeloos, W., & Van Gool, L. (2025). UniDepthV2: Universal monocular metric depth estimation made simpler. arXiv preprint arXiv:2502.20110.

[3] Höllein, L., Cao, A., Owens, A., Johnson, J., & Nießner, M. (2023). Text2room: Extracting textured 3d meshes from 2d text-to-image models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 7909-7920).

[4] Resolution-robust Large Mask Inpainting with Fourier Convolutions, Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, Victor Lempitsky Paper at arXiv 2109.07161

[5] 3D Gaussian Splatting for Real-Time Radiance Field Rendering, Bernhard Kerbl*, Georgios Kopanas*, Thomas Leimkühler, George Drettakis, Paper at arXiv 2308.04079

[6] T. Z. Xiang, G. S. Xia, X. Bai, and L. Zhang, "Image stitching by line-guided local warping with global similarity constraint," Pattern recognition, vol. 83, 481-497, 2018.

[7] T. Wu, C. Zheng, and T.-J. Cham, "PanoDiffusion: 360-degree panorama outpainting via diffusion," in International Conference on Learning Representations (ICLR), 2023.

[8] W. Yin, C. Zhang, H. Chen, Z. Cai, G. Yu, K. Wang, X. Chen, and C. Shen, "Metric3D: Towards zero-shot metric 3D prediction from a single image," in IEEE/CVF International Conference on Computer Vision (CVPR), 9043-9053, 2023.

[9] Christian Kerl, Jürgen Sturm, Daniel Cremers. Robust odometry estimation for rgb-d cameras. In ICRA, 2013.

[10] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. ACM Transactions on Graphics 2017 (TOG), 2017.

[11] Kshitij Goel and Wennie Tabib. Incremental multimodal surface mapping via self-organizing gaussian mixture models.IEEE Robotics and Automation Letters, 8(12):8358–8365,2023.

[12] Thomas Schops, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. InCVF/IEEE Conference on Computer Vision and PatternRecognition (CVPR), 2019.

[13] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, Olga Sorkine-Hornung. Differentiable surface splattingfor point-based geometry processing. ACM Transactions onGraphics (TOG), 38(6):1–14, 2019.

[14] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In CVPR, 2017.

[15] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J Davison. Scenecode: Monocular dense semanticreconstruction using learned encoded scene representations.In CVPR, 2019.

[16] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In ECCV, 2018.

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis. 3d gaussian splatting for real-timeradiance field rendering. ACM Transactions on Graphics, 42(4), 2023.

[18] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Trackingby persistent dynamic view synthesis. arXiv preprintarXiv:2308.09713, 2023.