Transforming SLAM Data into 3D Gaussian Splatting Models

Hansol Lim[1], Hanbeom Chang[1] and Jongseong Brad Choi[1*]

[1] Mechanical Engineering, State University of New York, Stony Brook, Stony Brook, NY 11794, USA

*corresponding Author

Abstract. Industrial robots equipped with cameras and LiDAR sensors generate vast amounts of data during regular operation, primarily for Simultaneous Localization and Mapping (SLAM) to navigate complex environments. However, SLAM-generated point clouds are often difficult for human experts to interpret. Despite containing rich and accurate environmental data, it remains underutilized for human visualization. In this paper, we demonstrate the use of 3D Gaussian Splatting (3DGS) to transform SLAM-generated point cloud data such as sparse LiDAR point cloud into highly detailed and visually appealing 3D model of the environment. The method offers realistic visualization opportunities that can be used in remote assessments by human experts, training grounds for reinforcement learning, and other various applications in visual analytics. Our results show that integration of LiDAR data from SLAM into 3DGS yields 1.218% and 9.186% increase in SSIM and PSNR, respectively. The results demonstrate that high quality 3D models of the environment can be achieved by using point clouds generated during SLAM, and can significantly enhance the performance of vanilla 3DGS models.

Keywords: 3D Gaussian Splatting, LiDAR, Sensor Fusion, SLAM

1 Introduction

Industrial robots today are increasingly equipped with both cameras and LiDAR sensors, enabling a multimodal sensing approach that allows them to perform Simultaneous Localization and Mapping (SLAM) with high precision in complex environments. By integrating these sensors, robots can fuse the rich visual data from cameras with the accurate depth measurements provided by LiDAR [1,2]. This combined data is processed into point cloud maps, which serve as the foundation for SLAM and optimal navigation. However, these SLAM-generated point cloud maps, while essential for navigation, are often difficult for humans to interpret due to their sparse and abstract nature [3].

Techniques such as 3D Gaussian Splatting (3DGS) have gained attention for their ability to render and reconstruct volumetric data more efficiently than traditional methods. 3DGS employs Gaussian functions to interpolate and project point clouds into continuous 3D representations, offering superior computational efficiency and model quality compared to conventional LiDAR point cloud meshing methods like Poisson Surface Reconstruction [4]. These advantages make 3DGS especially useful in fields requiring precise modelling and visualization [5].

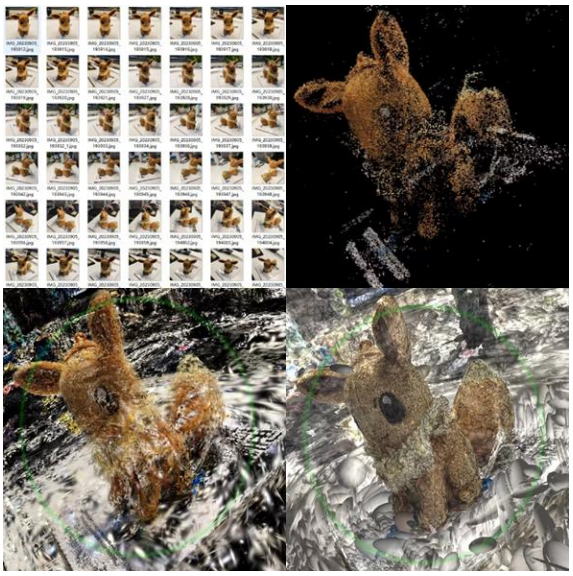

Fig. 1. Gaussian Splatting example taken from [6]

Despite its advantages, 3DGS faces challenges in handling sparse data and achieving high-resolution reconstructions. LiDAR sensing, known for its precise distance and depth measurements, can play a complementary role in overcoming these challenges. Integrating LiDAR data with Structure-from-Motion (SfM) and Neural Radiance Fields (NeRF) outputs has been shown to enhance the density and accuracy of 3D models. This integration is crucial in environments where optical methods are limited [7–10]. Future research directions may focus on optimizing these integrations for specific applications and exploring new computational methods to further enhance the fidelity and efficiency of 3D reconstructions [11].

The quality of 3DGS models opens up potential for enhancing the visual output of SLAM systems, making them more intuitive and informative for human operators. These models may also serve as valuable training environments for reinforcement learning algorithms, allowing robots to refine their navigation strategies before real-world deployment [12]. Moreover, detailed 3D maps generated by 3DGS significantly enhance remote sensing capabilities. Despite advances in SLAM

algorithms, robots are not always reliable, and remote diagnostics by human engineers are crucial for ensuring the safety of industrial operations [13]. By applying 3DGS, engineers can review the robot's SLAM-generated maps in a photorealistic 3D model and potentially identify obstacles or hazards that were not detected by the sensors.

In this paper, we explore the application of 3D Gaussian Splatting as a method for recycling SLAM data collected during robot operation into visually appealing 3D models. We investigate how the addition of LiDAR data can improve the quality of 3DGS models, addressing the challenges of sparse data and enhancing model accuracy. The results demonstrate that high-quality 3D model reconstruction can be achieved from recycling SLAM data, and that the integration of LiDAR data significantly improves the performance of vanilla 3DGS models.

2 Methodology

2.1 Sensor Calibration

To effectively fuse LiDAR and RGB camera data, precise calibration of the sensors was essential. This calibration ensured accurate alignment between the LiDAR point clouds and the corresponding visual information from the cameras, addressing the challenge of LiDAR's lack of color data [14–19].

We performed calibration by determining the relative transformation between the LiDAR and camera coordinate systems. This transformation is represented by a rotation matrix R_{LC} and a translation vector t_{LC} calculated using checkerboard calibration tests where known patterns allowed accurate computation of transformation parameters.

The transformation from the LiDAR coordinate system P_L to the camera coordinate system P_C is defined as:

$$P_C = R_{LC}P_L + t_{LC}$$

Values for R_{LC} and t_{LC} were iteratively derived to minimize alignment errors. This process aligns with sensor fusion techniques that enhance recognition capabilities by combining LiDAR data with RGB images, thus imbuing point clouds with visual information. By tightly coupling the camera-LiDAR system and incorporating loop-closure modules, we minimized drift in the final point cloud output, resulting in a more accurate visual map for inspection.

To generate colored point clouds, we utilized R3LIVE (Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled State Estimation and mapping package) [20, 21,22]. This system integrates LiDAR, camera, and inertial measurement unit (IMU) data to improve state estimation accuracy.

The key method employed is the Error-State Iterated Kalman Filter (ESIKF), which compensates for motion distortion in LiDAR data by backward propagating the IMU data. The state error $\delta\hat{x}_k$ is estimated by linearizing the system dynamics and sensor measurements around the nominal state.

$$\delta\overline{\hat{x}_k} = F_k\delta\hat{x}_{k-1} + G_k w_{k-1}$$

Where F_k is the state transition matrix, G_k is the process noise matrix, and w_{k-1} is the process noise.

The update step with LiDAR and IMU measurements:

$$K_k = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$$

$$\delta\hat{x}_k = \delta\overline{\hat{x}_k} + K_k(z_k - H_k\delta\overline{\hat{x}_k})$$

$$P_k = (I - K_k H_k)\overline{P_k}$$

Where K_k is the Kalman gain, P_k is the error covariance matrix, H_k is the measurement matrix, R_k is the measurement noise covariance, and z is the measurement vector. This refined state estimation allowed for the accurate addition of points from each LiDAR scan to the global 3D point cloud, enhancing the density and accuracy of the model.

Following sensor calibration, we employed COLMAP SfM to reconstruct a 3D structural point cloud using image data. SfM reconstructs 3D structures from 2D image sequences by estimating camera poses and scene geometry.

The Fundamental Matrix F encapsulates the intrinsic projective geometry between two views, relating corresponding points x_i and x'_i

$$x_i^{T\prime} F x_i = 0$$

To refine the relationship, the Essential Matrix E incorporates internal camera calibration matrices K and K':

$$E = K'^T F K$$

E is decomposed to extract rotation R and translation t between camera views:

$$E = [t]_{\times} R$$

Bundle adjustment optimizes camera parameters C_j amd 3D points P_i by minimizing the reprojection error:

$$min_{C_j,P_i} \sum_{i,j} \left\| x_{ij} - \Pi(C_j, P_i) \right\|^2$$

Here, C_j represents camera parameter (including position and orientation), Π denotes the coordinates of the 3D point in the scene x_{ij} are the observed projections of point Π in image j, and Π is the projection function mapping 3D points to their 2D image counterparts in C_j.

2.2 Data Collection

2.2.1 Equipment and Environment set-up

Data was collected using a custom-designed rover equipped with the following sensors:

Table 1. Equipment specifications

CPU	Intel i7-13000KF	
M/B	ASRock Z790 PG Lightning	
GPU	NVIDIA RTX 4080 16GB	
RAM	Crucial DDR5 64GB	
SSD	SAMSUNG 990 Pro 1TB	
	**Camera: FLIR Blackfly S Specifications**	
	Resolution	1440 x 1080
	Frame Rate	60 FPS
	Field of View	70°
	Camera Sensor	SONY IMX273
	CMOS Sensor Size	9 ½”
	**LiDAR: Ouster OS0-32 Specifications**	
	Horizontal Scan	360°
	Vertical Scan	90°
	Maximum Range	35m
	Resolution	1024 x 24 Hz
	Data Transfer	655,360 points/s

The rover navigated a complex indoor environment resembling a ship's engine room (Figure 3). The primary objective was to perform SLAM while capturing both geometric and visual data.

Fig. 2. Rover and environment used for dataset generation

The data for this study was gathered using a custom-designed rover equipped with a LiDAR sensor and an RGB camera system. These sensors were mounted to capture both geometric and visual information as the rover navigated a complex indoor ship engine environment. The primary objective of the rover's navigation was to SLAM within this environment. The data collected during this SLAM operation was subsequently repurposed for the generation of high-fidelity 3D maps using 3DGS.

The LiDAR sensor captured a dense point cloud representing the spatial geometry of the environment, while the RGB camera recorded high-resolution images of the scene. The combination of these two data streams enabled us to gather a comprehensive dataset suitable for both localization and detailed environmental modeling.

The position p_i of each point i in the point cloud was defined by its 3D coordinates:

$$p_i = (x_i, y_i, z_i)$$

Where $x_i, y_i,$ and z_i represent the spatial position of the point relative to the rover. Simultaneously, the RGB camera captured color data $c_i = (r_i, g_i, b_i)$ where $r_i, g_i,$ and b_i represent the red, green, and blue channels of the color associated with each point in the 3D environment.

To ensure accurate capture, the rover was equipped with IMU that provided orientation and velocity data, enabling us to estimate the motion of the rover during data acquisition. The LiDAR system emitted laser pulses that measured the time-of-flight of each pulse to calculate the distance to various objects. These

measurements formed the basis of the point cloud, while the IMU data was used to correct for any motion-induced distortions.

$$d_i = \frac{1}{2} c t_i$$

Where d_i is the distance to the object, c is the speed of light, and t_i is the time-of-flight of the LiDAR pulse. The RGB data was synchronized with the LiDAR measurements to ensure that each 3D point was associated with the correct color information from the camera.

2.3 Sensor Data Fusion

To align the data from the LiDAR and camera sensors, we used CloudCompare for manual registration. First, the point cloud from the LiDAR sensor was loaded into the software. Then, the corresponding camera images were imported, and initial alignment was done manually by matching key landmarks visible in both the LiDAR and camera data. Following this manual alignment, the Iterative Closest Point (ICP) algorithm was applied to fine-tune the registration, minimizing the error between the 3D point cloud and the 2D image projections.

ICP algorithm can be used to fuse two point clouds from by minimizing the following equation:

$$\min_{R,t} \sum_i \|R p_i + t - q_i\|^2$$

Where R is the rotation matrix, t is the translation vector, p_i represents the points from the LiDAR dataset, and q_i are the corresponding points from the camera dataset. This step ensured that the depth and color information were accurately combined, resulting in a well-aligned multimodal dataset.

2.4 3D Gaussian Splatting Training

After fusing the LiDAR and camera data, we applied the 3D Gaussian Splatting (3DGS) technique to process the resulting point cloud. This approach models each point as a Gaussian splat, transforming the discrete point cloud into a continuous volumetric representation.

Each point p_i in the point cloud was transformed into a Gaussian with mean $\mu_i = (x_i, y_i, z_i)$ and covariance matrix σ_i representing the uncertain in the position of the point:

$$G_i(p) = \frac{1}{(2\pi)^{3/2}|\Sigma_i|^{1/2}} exp\ (-\frac{1}{2}(p-u_i)^T \Sigma_i^{-1}(p-u_i)$$

Where x is a position in 3D space, μ is the 3D positioning of the point being represented, and Σ is the covariance matrix.

Optimization was performed using gradient descent methods over 30k iterations. The inclusion of LiDAR data in the 3DGS model provided additional depth information, enhancing the density and accuracy of the reconstructed 3D models.

Fig. 2. Loss, PSNR, SSIM vs iteration

2.5 Model Performance Evaluation

To evaluate the quality and accuracy of the 3D models generated by 3D Gaussian Splatting, we used two primary metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

PSNR is a commonly used metric to evaluate the quality of a 3D model or image by comparing the similarity between the reconstructed model and the ground truth. It measures the ratio between the maximum possible value of the signal and the noise affecting its representation. A higher PSNR value indicates a better-quality model with fewer distortions. PSNR is computed as:

$$PSNR = 10 \log\left(\frac{MAX_I^2}{MSE}\right)$$

Where is MAX_I the maximum possible intensity value of the image or model, and MSE is the Mean Squared Error between the ground truth and the reconstructed model.

SSIM is used to assess the perceived quality of the 3D model by evaluating the structural similarity between the model and the reference (ground truth). It compares luminance, contrast, and structure between two images or models to provide a more comprehensive measure of perceptual quality.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Where μ_x and μ_y are the means of images x and y, σ_x^2 and σ_y^2 are the variances, and σ_{xy} is the covariance of x and y. The constants C_1 and C_2 are used to stabilize the division.

3 Results

3.1 Performance Assessment

Model generated with SLAM data was able to generate a high-fidelity model. Moreover, it demonstrated superior performance across all metrics compared to the vanilla model that only used camera data. SSIM increased by 0.011575 (+1.218%), and PSNR increased by a significant margin of 2.0477 (+9.196%). This is a significant improvement visual and textural fidelity when compared to camera-only 3DGS models.

Table 2. Performance Metric Assessment

Performance at 30k iterations			
	Loss	SSIM	PSNR
Vanilla	0.022443	0.950478	22.29116
Reinforced	0.019329	0.962053	24.33886
Performance	-13.87%	+1.218%	+9.186%

Fig.3. Ground Truth (Left), SLAM data 3DGS model (Right)

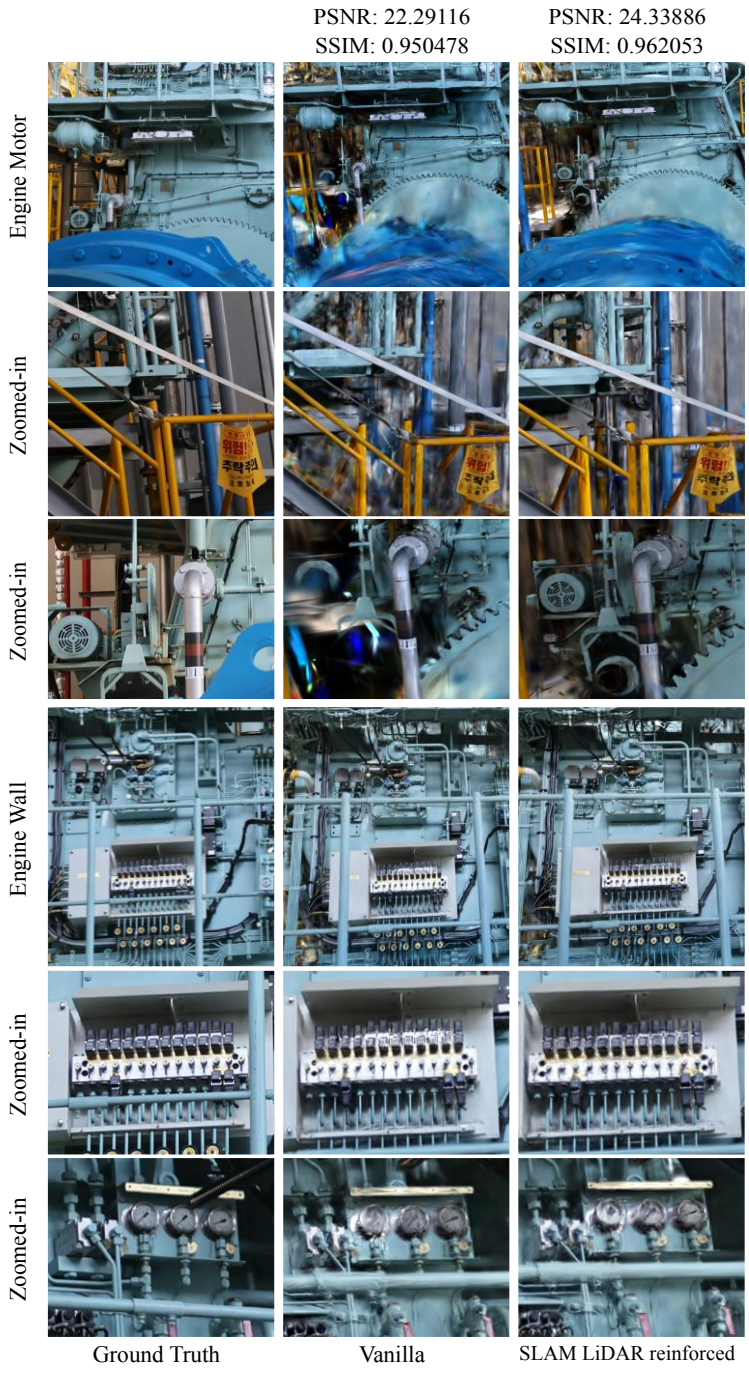

Fig.4. Qualitative asessment of various details of the ship engine

The performance of the 3DGS model was evaluated after 30k training iterations using the vanilla model and model reinforced with SLAM LiDAR data, with the key metrics being Loss, SSIM, and PSNR:

4 Conclusion

In this study, we demonstrated that the vast amounts of SLAM data already being collected by robots can be recycled into a high-quality 3D model using 3D Gaussian Splatting. The generated models offer a more intuitive, photorealistic view of the robot's workspace. This allows engineers to visually assess environments, identify potential obstacles, and detect hazards that may have been missed by robot sensors. By integrating LiDAR data, we also addressed challenges related to 3DGS's weaknesses on camera-only rendering and enhanced model quality proven by the significant increase in PSNR and SSIM values. This approach shows that the large volumes of SLAM data produced by current robots can be repurposed for 3D Gaussian Splatting. This advancement extends the utility of SLAM-generated data and opens new opportunities for industrial applications that require visually appealing environment maps for human experts.

References:

[1] Guo, N., Ma, X., Liu, G., & Feng, Z. (2024). RLI-SLAM: Fast Robust Ranging-LiDAR-Inertial Tightly-Coupled Localization and Mapping. *Sensors*, 24(17), 5672. https://doi.org/10.3390/s24175672

[2] Bosch Research. (2023). Simultaneous localization and mapping. Retrieved from https://www.bosch.com/research/slam

[3] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Fields Rendering," ACM Transactions on Graphics, vol. 42, no. 4, pp. 1-22, 2023.

[4] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson Surface Reconstruction," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, 2006, pp. 61-70.

[5] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds, "'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications," Geomorphology, vol. 179, pp. 300-314, 2012.

[6] "Exploring 3D Gaussian Splatting," Bilibili, 2021. [Online]. Available: www.bilibili.com/read/cv26465887/.

[7] W. Zhen, Y. Hu, H. Yu and S. Scherer, "LiDAR-enhanced Structure-from-Motion," in 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 6773-6779, doi: 10.1109/ICRA40945.2020.9197030.

[8] T. Tao, et al., "LiDAR-NeRF: Novel lidar view synthesis via neural radiance fields," arXiv preprint, 2023. Available: https://arxiv.org/abs/2304.10406.

[9] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in Proceedings of the European Conference on Computer Vision (ECCV), 2020, pp. 405-421.

[10] J. Chen and B. Bhanu, "3D free-form object recognition in range images using 3D Gaussian Splatting," Pattern Recognition Letters, vol. 131, pp. 355-362, 2020.
[11] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5828-5839.
[12] Flyability. (2023). Understanding SLAM in Robotics and Autonomous Vehicles. Retrieved from https://www.flyability.com/slam-in-robotics
[13] Ouster. (2023). Introduction to SLAM (Simultaneous Localization and Mapping). Retrieved from https://ouster.com/blog/what-is-slam
[14] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R2 live: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping," IEEE Robotics and Automation Letters, vol. 6, no.4, pp. 7469–7476, 2021.
[15] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," arXiv preprint arXiv:2104.10831, 2021.
[16] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "Lic-fusion: Lidarinertial-camera odometry," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019, pp. 5848–5854.
[17] X. Zuo, Y. Yang, J. Lv, Y. Liu, G. Huang, and M. Pollefeys, "Lic-fusion 2.0: Lidar-inertial camera odometry with sliding-window plane-feature tracking," in IROS 2020, 2020.
[18] J. Lin and F. Zhang, "R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial Visual tightly-coupled State Estimation and mapping package," arXiv preprint, 2021. Available: https://arxiv.org/abs/2109.07982.
[19] J. Lin and F. Zhang, "R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package," in 2022 International Conference on Robotics and Automation (ICRA), 2022.
[20] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense Tracking and Mapping in Real-Time," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2320-2327.
[21] ROS Wiki, "Camera Calibration," 2023. [Online]. Available: https://wiki.ros.org/camera_calibration.
[22] K. Koide, "Direct Visual-LiDAR Calibration," GitHub Repository, 2023. [Online]. Available: https://github.com/koide3/direct_visual_lidar_calibration.