IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Human-in-the-Loop Gaussian Splatting for Robotic Teleoperation

Yongseok Lee, Hyunsu Kim, Harim Ji, Jinuk Heo, Youngseon Lee, Jiseock Kang, Jeongseob Lee, *Member, IEEE*, and Dongjun Lee, *Member, IEEE*

***Abstract*—Safe, precise teleoperation demands a third-person 3D view that reveals collision clearances and task-critical geometry in full detail. Yet most systems still rely on live camera streams that offer tunnel-vision perspectives and weak depth cues, hiding hazards and denying operators the spatial context for precise manipulation. 3D Gaussian Splatting (GS) renders photorealistic views in real time, yet safe, efficient multi-view acquisition in cluttered teleoperation remains a bottleneck. We propose Human-in-the-Loop Gaussian Splatting (HIL-GS) that delivers safe, robust, and efficient 3D scene reconstruction for challenging teleoperation environments. HIL-GS combines three modules in a tightly-coupled loop: (1) motion-aware GS reconstruction that fuses RGB-D and proprioceptive sensors for drift-free and robust mapping under aggressive motions; (2) VR-based informative display that renders the GS map with contextual overlays/feedback in real time to ensure situational awareness and reconstruction completeness; and (3) finger-based control interface to guide the robot toward informative viewpoints through safe, non-redundant motions. Through simulation and real-world experiments, we demonstrate that HIL-GS outperforms traditional approaches in reconstruction quality, usability, and efficiency.**

***Index Terms*—Telerobotics, simultaneous localization and mapping, human in the loop, user interfaces, digital twins, sensor fusion.**

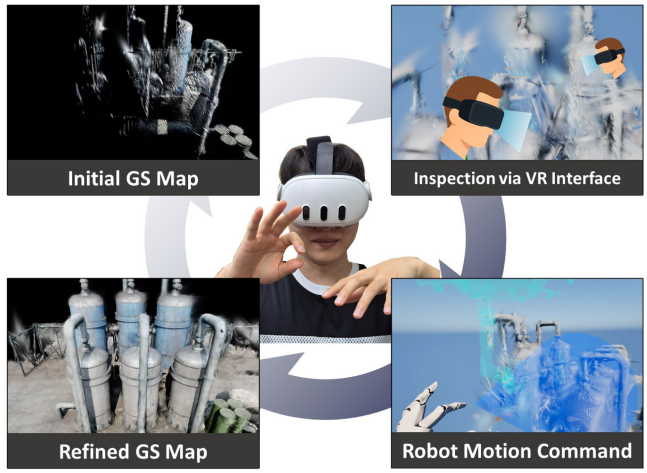

Fig. 1. Overview of our HIL-GS framework: An initial GS map is generated from early robot observations. The human operator inspects the 3D scene via a VR interface and, using a predictive display, selects safe and efficient robot motion commands. The robot then executes the motion, gathers new observations, and refines the map. .

I. Introduction

TELEOPERATION allows humans to operate in hazardous environments (radioactive, high-elevation, or disaster-stricken site). However, conventional video-based teleoperation, which streams live imagery from onboard cameras, suffers from a critical limitation despite its straightforward implementation. It provides only fragmentary situational awareness, as its narrow field of view and ambiguous depth cues result in poor depth perception, hidden occlusions, and a loss of structural context [1]. This limitation is exacerbated by the camera's fixed position on the robot body. Without the ability to freely change their viewpoint (e.g., using pan-tilt gimbals), operators cannot inspect critical details—such as obstacles at the edge of the view or the occluded backside of a target object—which ultimately hinders 3D understanding and prevents safe, precise manipulation [2].

Reconstruction-based approaches provide a promising alternative by modeling the remote scene in 3D. Recent radiance-field methods—most notably Gaussian Splatting (GS) [3]—render photorealistic views at interactive rates, giving the operator far richer spatial context for inspecting occlusions and fine geometry [4]. However, high-quality GS requires images from many diverse viewpoints. Collecting such views in cluttered teleoperation settings remains a significant challenge, as existing autonomous planners often fail to generate safe and efficient paths in real time. While methods like active sensing [5] can find informative viewpoints and constrained motion planning [6] can ensure safety, robustly integrating both in complex, non-convex or occluded environments is often intractable. Moreover, the aggressive or rotation-only motions typical of teleoperation, combined with challenging visual scenes such as extremely near or distant backgrounds, can cause vision-only pipelines to drift or fail [4], [5].

To address these challenges, we propose Human-in-the-Loop Gaussian Splatting (HIL-GS) teleoperation framework that integrates robust 3DGS reconstruction, VR informative display, and intuitive finger-based control. As illustrated in Fig. 1, HIL-GS

Received 20 July 2025; accepted 5 November 2025. Date of publication 14 November 2025; date of current version 19 November 2025. This article was recommended for publication by Associate Editor Michael Hagenow and Editor Ki-Uk Kyung upon evaluation of the reviewers' comments. This work was supported in part by Ministry of Trade, Industry & Energy (MOTIE, Korea) under Grant RS-2024-00441872, and in part by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) under Grant RS-2025-00521109. *(Corresponding author: Dongjun Lee.)*

The authors are with the Department of Mechanical Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: djlee@snu.ac.kr).

Digital Object Identifier 10.1109/LRA.2025.3632755

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

operates in a tight, iterative loop: (1) The robot generates an initial coarse map from its observations. (2) To improve the reconstruction quality of a target, the operator inspects this map in VR and uses a predictive display to select a safe and informative next viewpoint. (3) The robot then executes the motion, captures new data, and refines the map. This iterative process allows the operator to incrementally build a complete and accurate 3D scene, ensuring safe and efficient teleoperation.

The main contributions of this work are as follows:

1) **Motion-aware GS reconstruction:** Fuses RGB-D imagery with proprioceptive sensors (IMU, encoders) to produce robust, drift-free, high-fidelity maps.
2) **VR-based informative display:** Streams the live GS map with informative overlays (unobserved-region (voxels), camera frustums, predicted robot pose, and collision-risk warning) via a stereoscopic VR interface.
3) **Finger-based control interface:** Employs hand tracking for interactive next-viewpoint (robot pose) selection and reconstructed-map inspection, closing the loop of online reconstruction (Fig. 1).

Compared with conventional video–stream teleoperation or fully autonomous navigation, our simultaneous reconstruction and teleoperation interface offers three advantages: (i) **efficiency**—expert/semantic priors guide viewpoints to task-critical structure, accelerating reconstruction quality with fewer views and shorter paths; (ii) **safety**— display of the GS map with VR overlays/feedback helps operators anticipate hazards and choose safer motions during operation; and (iii) **robustness**—sensor fusion of RGB-D and proprioception prevents SLAM failures under aggressive motions or in challenging scenes.

II. Related Works

A. Reconstruction-Based Visual Feedback for Teleoperation

Reconstruction-based visual feedback has gained prominence to overcome the narrow viewpoint and weak depth perception of camera-stream teleoperation [1], [2]. Early work used SLAM pipelines to build sparse feature maps or volumetric models (e.g., OctoMap and Voxblox) to supply basic 3D context [7], [8]. Although real-time, these maps lack photorealistic detail and are prone to drift, especially in large, cluttered scenes.

Radiance-field methods such as Neural Radiance Fields (NeRF) [9] provide highly photorealistic, view-dependent scenes. For instance, [10] proposed a NeRF-based predictive display for robotic teleoperation, and [11] applied NeRF to remote collaboration. However, implicit NeRF models require minutes of offline optimisation and render each new view in several seconds, limiting their use in dynamic, real-time teleoperation. 3D Gaussian Splatting (3DGS) is a recent explicit alternative that matches NeRF's visual quality yet renders orders of magnitude faster [3], thus utilized for robotic teleoperation [4]. Even so, dense 3DGS still needs many multi-view images — hard to collect when a teleoperated robot carries only a forward camera and must follow cautious, short trajectories [5].

Recent works have begun integrating GS with VR. VR-Splat [12], for instance, refined the pipeline for more effective VR rendering, while VR-GS [13] converts Gaussians to meshes for physical simulation. However, these systems primarily focus on visualizing and interacting with static or pre-existing scenes; they do not address the key challenge of reconstructing the map online within a real-time teleoperation loop. To bridge this gap, our HIL-GS framework allows an operator to select task-critical viewpoints via a finger-tracked VR interface, with our motion-aware GS mapping continuously updating the 3D map with every robot move.

B. Human-in-The-Loop Reconstruction and Mapping

Human-in-the-loop (HIL) reconstruction exploits operator insight to correct drift, add semantics, and focus sensing where autonomy may fail, in dynamic or cluttered scenes. Early systems such as A-SLAM [14] let users refine a 2-D occupancy grid through hand gestures in an AR headset, while [15] provided a GUI for adjusting object placement and relationships. [16] made the loop tighter, allowing point-cloud alignment in real time via mouse input in RViz.

In teleoperation, HIL has been used to mark regions of interest and hazards: [17] inserted operator labels for semantic mapping, and [18] let users flag dangerous zones on a live camera feed. More recent work pursues richer scene understanding. [19] proposed HSS-SLAM, where operators refine superquadric object models on-line, and [20] studied multi-user mapping for shared situational awareness. Most of these systems rely on point clouds or voxel grids and accept only coarse mouse/keyboard input, offering limited visual realism and dexterity. None integrates modern radiance-field maps such as NERF or 3DGS.

In summary, our work adopts the HIL paradigm as it is better suited for teleoperation in large-scale cluttered environments than autonomous alternatives. Fully autonomous planners, for instance, lack the semantic understanding to prioritize mission-critical targets, while common "plan-then-refine" semi-autonomous pipelines lack the continuous, real-time expert intervention needed in hazardous, unknown scenes. Our framework leverages these core HIL strengths and significantly advances prior HIL systems by replacing their coarse or non-photorealistic maps with live 3DGS and trading keyboard-mouse inputs for an intuitive, finger-based VR interface. To our knowledge, ours is the first system to couple these features with predictive configuration previews and map-aware collision warnings into a complete and robust teleoperation solution.

III. Methods

Fig. 2 summarizes the HIL-GS framework, which combines motion-aware GS reconstruction, VR display, and finger-based control in a closed human–robot loop.

A. Motion-Aware GS Reconstruction

Accurate 3DGS reconstruction for teleoperation is challenging, as existing GS research focuses mainly on static datasets [21] or controlled indoor cameras [22], [23] with little attention to real-world applications. These teleoperation scenarios introduce a combination of agile robot motions, visually challenging scenes, and network limitations that cause GS reconstruction to become unstable or fail.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

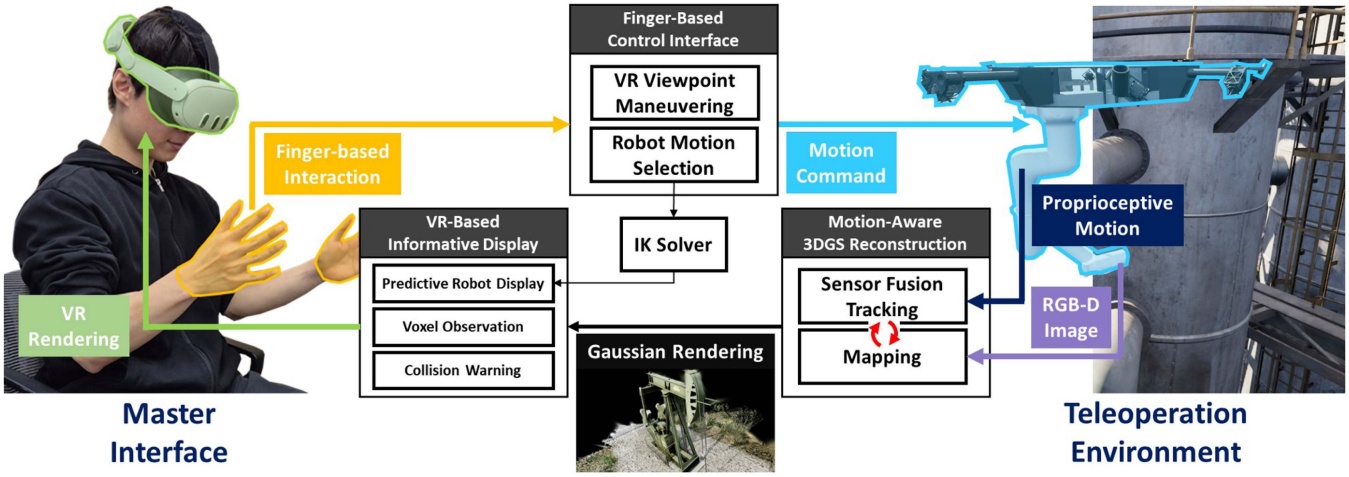

Fig. 2. A descriptive pipeline of our framework: In the teleoperation environment, the robot-mounted camera transmits RGB-D images, while proprioceptive sensors (e.g., IMU, encoders) provide robot motion information. The motion-aware 3DGS reconstruction module fuses these sensor data to construct an accurate and robust GS Map. This GS Map is rendered in VR and transmitted to the operator, who observes it through a VR interface and provides real-time feedback using a finger-based interaction.

To overcome these issues, we augment a state-of-the-art GS-ICP SLAM pipeline [21] with proprioceptive robot sensors (e.g., IMU, joint encoders). GS-ICP SLAM employs *Generalized-ICP (G-ICP)* to rapidly align point clouds sampled from a depth camera, enabling real-time Gaussian mapping. We further extend this algorithm with two additional capabilities: (i) a keyframe selection, where the original system relied on heuristics such as overlap between point clouds or fixed frame intervals, but our modified pipeline leverages proprioceptive tracking information to compare against the current frame and generate keyframes more robustly; and (ii) a depth masking, which prevents Gaussian generation in regions with missing depth values (e.g., points that are too close or too far from the sensor), thereby enhancing robustness in outdoor environments where the original algorithm was limited to indoor use.

Our fusion method integrates proprioceptive data at two stages: (1) refining the initial pose guess for G-ICP and (2) performing an adaptive fusion with the G-ICP result. This two-stage approach, which seeds the registration with motion priors, keeps the pipeline stable under challenging conditions.

Let $T_{\text{prev}} \in SE(3)$ be the previous pose and $T_{\text{prop}} \in SE(3)$ the current proprioceptive pose. The pose change in Lie algebra is:

$$\Delta\xi_{\text{prop}} = \log(T_{\text{prev}}^{-1} T_{\text{prop}}), \tag{1}$$

where $\Delta\xi_{\text{prop}} \in \mathbb{R}^6$ represents the logarithmic map of the relative pose change between T_{prev} and T_{prop}. We then transform the target point cloud using T_{prop}:

$$X_{t^*} = exp(\Delta\xi_{\text{prop}}) \cdot X_t, \tag{2}$$

and use X_{t^*} as the predicted target point cloud for G-ICP to improve the correspondence search by providing a more accurate starting point.

The cost function is then minimized to find the transformation:

$$T_{\text{icp}} = \underset{T \in SE(3)}{\arg\min}\, J_{\text{icp}}(T), \tag{3}$$

where

$$J_{\text{icp}}(T) = \sum_{i=1}^{N} \left(x_i^t - T\, x_i^s\right)^{\top} \left(C_i^t + T\, C_i^s\, T^{\top}\right)^{-1} \left(x_i^t - T\, x_i^s\right), \tag{4}$$

with X_s and $X_{t_{\text{cur}}}$ representing the source and transformed target point clouds, respectively, $x_i^s \in X_s$ and $x_i^t \in X_{t_{\text{cur}}}$ the i-th corresponding points between them, N the total number of correspondences, and C_i^s, C_i^t their associated covariance matrices.

After obtaining T_{icp}, we compute the discrepancy between T_{icp} and T_{prop} using the logarithmic map:

$$\Delta\xi = \text{Log}\left(T_{\text{icp}}^{-1} T_{\text{prop}}\right), \tag{5}$$

which yields a 6-dimensional vector decomposed into a translational error $\Delta\xi_t \in \mathbb{R}^3$ and a rotational error $\Delta\xi_R \in \mathbb{R}^3$. The uncertainty in the sensor measurements is modeled by:

$$\Sigma_{\text{prop}} = \begin{bmatrix} \Sigma_{\text{t}} & 0 \\ 0 & \Sigma_{\text{R}} \end{bmatrix}, \tag{6}$$

where $\Sigma_{\text{t}} \in \mathbb{R}^{3\times3}$ and $\Sigma_{\text{R}} \in \mathbb{R}^{3\times3}$ represent the translational and rotational covariances, respectively. To adaptively reweight the rotational term in near-pure rotation, we define:

$$\lambda_R = \frac{\alpha}{\|\Delta\xi_t\| + \varepsilon}, \tag{7}$$

where α is a positive scaling constant and ε prevents division by zero. For large-distance effects, we define D as the average distance to observed points and adjust the sensor prior weight:

$$\lambda = \lambda_0 \times \exp\left(\beta\, D\right), \tag{8}$$

where λ_0 is a base weight and β controls the dependence on distance. The final pose is refined as:

$$T^* = T_{\text{icp}} \cdot \text{Exp}\left(\lambda \begin{bmatrix} \Sigma_{\text{t}}^{-1} \Delta\xi_t \\ \lambda_R \Sigma_{\text{R}}^{-1} \Delta\xi_R \end{bmatrix}\right). \tag{9}$$

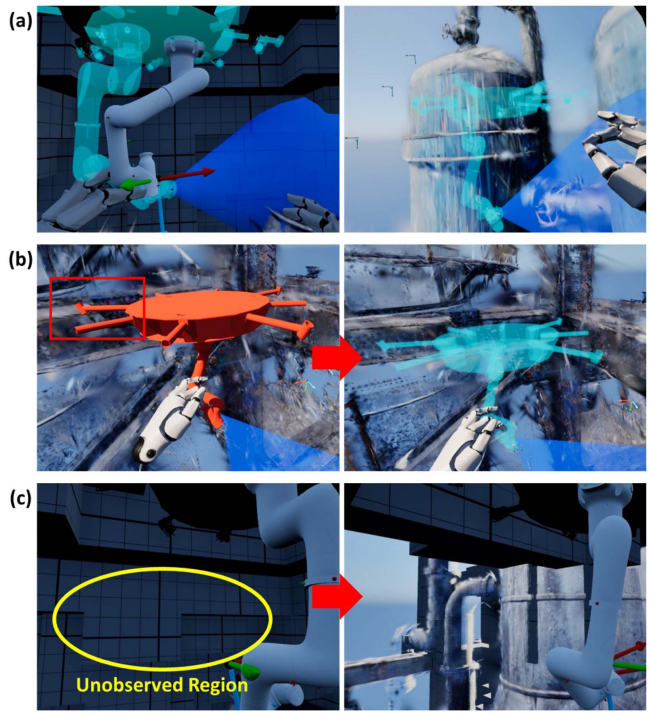

Fig. 3. (a) Predictive robot motion and camera FOV semi-transparent display. (b) When a potential collision is detected (red box), the predictive robot motion turns red to alert the user, allowing them to adjust the motion to avoid unsafe behavior. (c) Initially unobserved regions (gray voxels) are either removed or replaced by 3D Gaussians once observed. .

Incorporating proprioceptive data in both initial alignment and adaptive fusion enhances robustness. The initial adjustment improves pose estimates under rotation-only or fast motion or visually-challenging scenes where vision-only matching fails (e.g., distant backgrounds), while adaptive fusion combines visual and motion cues for reliable, accurate reconstruction across diverse conditions.

B. VR-Based Informative Display

For safe and effective teleoperation, we develop a VR-based informative display that visualizes live 3DGS reconstruction and predictive overlays to support immersive inspection and expert-in-the-loop reconstruction (Fig. 3).

1) Implementation Setup: To distribute the computational burden, the system consists of two PCs (Intel i9-13900 K/Ryzen 9900x, 32 GB RAM, NVIDIA RTX 4090): a GS-PC and an Interface-PC. The GS-PC receives sensory data (RGBD, encoder) from the remote robot and runs the entire GS-SLAM pipeline. The Interface-PC receives the resulting GS map data from the GS-PC, processes the operator's gesture inputs, and renders the VR scene with overlays (e.g., predictive previews, collision warnings). The Meta Quest 3 headset serves only as the primary display and is tethered to the Interface-PC via a wired link. We implement 3DGS rendering system by adapting LumaAI's particle renderer plugin [24] and customizing it for real-time updates of large-scale GS maps with up to two million Gaussian splats.

The PCs and the remote robot communicate using the MQTT protocol, handling three primary data streams: (1) Robot to GS-PC: A raw sensor feed, including 1280 $\times$ 720 RGB-D images, is streamed at 2Hz, requiring approximately 110Mbps of bandwidth. (2) GS-PC to Interface-PC: To overcome the key challenge of transmitting the ever-growing GS map, our optimized pipeline streams only incremental updates. This sustains 10,000 Gaussian splat updates at a rate of 2 Hz, utilizing a bandwidth of approximately 7 Mbps. (3) Interface-PC $\leftrightarrow$ Robot: The main control loop for sending motion commands and receiving pose feedback runs at a responsive 20Hz, requiring minimal bandwidth.

2) Voxel Observation Display: Unobserved areas around the robot or target objects may hide unmodeled obstacles or posing safety risks if traversed blindly. We render a voxel-based observation map around the robot and target objects. The workspace (typically a 10 m radius at 0.5 m resolution) is discretized into voxels initialized as unobserved (gray); a voxel flips to observed (transparent) when seen from valid viewpoints—depths within range and not occluded by existing Gaussians.

3) Predictive Robot Display: To designate next motion safely, the VR interface previews the robot at the operator's hand-guided next configuration. When the operator performs a right-hand pinch, the end-effector is constrained to the right-hand pose; the ghost (predicted robot configuration and camera FOV) follows the fingertip in real time by solving IK with MoveIt [25]. The execution workflow is described in Section III-C3.

4) Collision Warning: Our interface is integrated with real-time collision checks between the predictive robot display and the constructed GS map to ensure a collision-free path. We wrap each robot link and Gaussian splat with a conservative ellipsoid, making the collision checks between the display and the map a series of overlap tests between pairs of ellipsoids. Suppose, a pair of ellipsoids are defined as a pair of tuples; $\{p_i,\ R_i,\ U_i\}_{i=1,2}$, where $p_i \in \mathbb{R}^3$ is the position of the center, $R_i \in SO(3)$ is the rotation, and $U_i \in \mathbb{R}^3$ is the scale of the ellipsoid. Based on the separating axis theorem, the test can be done by predicating whether the maximum value of the function defined in the boundary of the unit ball, $G : \partial\mathcal{B}(0,1) \rightarrow \mathbb{R}$, which is defined as follows:

$$G(d) = c \cdot d - \|S^T d\| \tag{10}$$

where $c = U_2^{-1} R_2^{-1}(p_1 - p_2)$, $S = U_2^{-1} R_2^{-1} R_1 U_1$ is more than 1. Although this test can be performed using a simple projected gradient method, it requires several million tests for a single collision check. To deliver the collision warning to the user at an interactive rate, we parallelize the collision check at each test level with a GPU, achieving sub-millisecond computation times. As shown in Fig. 3(b), the predictive robot display turns red right after the collision check reports a collision between the display and the map, leading the user to select other safe paths.

C. Finger-Based Control Interface

1) Hand Tracking: Our teleoperation framework integrates an intuitive finger-based interface for intuitive viewpoint control in VR. We adopt a modular structure compatible with diverse

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

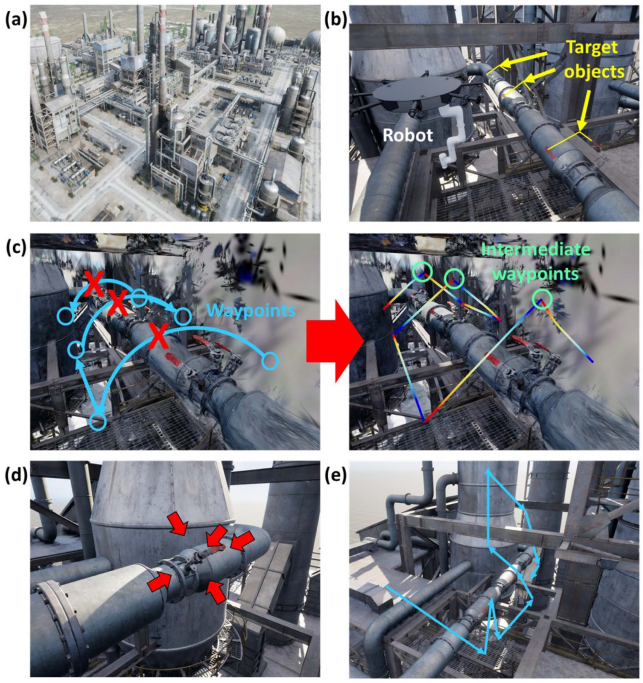

Fig. 4. (a) Large-scale refinery VR environment. (b) Three target valves and our teleoperated robotic platform. (c) Trajectory of Splat-Nav baseline, which was given a pre-built GS map and required manual intermediate waypoints to handle planner failures. (d) Ground-truth views for evaluation per valve. (e) Representative HIL-GS trajectory guided by live feedback. .

finger-tracking devices. We tested two systems: a wearable Visual-Inertial Skeletal Tracking (VIST) system [26] providing high-accuracy tracking suitable for demanding scenarios and haptic integration, and the built-in hand tracking of Meta Quest 3, which offers natural gesture interaction without additional hardware.

2) VR Viewpoint Maneuvering: Operators maneuver the viewpoint within the reconstructed 3DGS environment using natural hand gestures. Right-hand pinch–drag translates the view, forward/backward motion zooms by changing camera distance, and two-handed pinch–twist rotates the view, enabling full 6-DoF reorientation. Combined with 3DGS VR rendering, this interface allows precise inspection of areas needing detailed reconstruction and assessment of potential safety risks — both critical for intricate teleoperation tasks.

3) Robot Motion Selection: As described in Section III-B, operators use a right-hand pinch gesture to preview the ghost (predictive robot configuration and camera FOV) while receiving collision warnings with the environment in real-time. To designate the next robot motion, a left-hand pinch fixes the ghost, after which the operator evaluates suitability with respect to the surrounding layout, joint limits/singularities, and potential self/scene occlusions during the transitional motion. After inspecting the prospective pose by the above viewpoint maneuvering, the user either executes the motion (right-hand pinch) or cancels (left-hand pinch) and continues exploring alternative poses. This multi-stage process of preview and evaluation enables operators to leverage their expertise, identifying informative viewpoints and avoiding redundant viewpoints while ensuring safety and efficiency.

IV. Simulation Experiment

A. Experimental Setup

We created a photorealistic virtual oil-refinery in Unreal Engine (Fig. 4) that captures dense, occlusion-rich geometries such as pipelines, machinery and valves, mirroring the visual challenges of industrial sites. The robot platform for teleoperation is a Suspended Aerial Manipulator (SAM) [27], comprising a drone equipped with a 6-DOF manipulator arm. Reconstruction quality is evaluated against ground-truth images using three standard photometric metrics. We report Peak Signal-to-Noise Ratio (PSNR) to measure pixel-level accuracy, Structural Similarity Index Measure (SSIM) to assess perceptual similarity in structure and contrast, and Learned Perceptual Image Patch Similarity (LPIPS), which uses deep features to closely align with human judgment of image quality. For PSNR and SSIM, higher scores indicate better reconstruction fidelity, whereas for LPIPS, a lower score is better as it represents a smaller perceptual distance.

B. Experiment 1

1) Objective and Baseline: We test whether HIL-GS yields more efficient and safer trajectories for reconstruction of the target object (three valves on a pipe) than an autonomous pipeline. Instead of map-free active-view methods (e.g., ActiveSplat [5]), which assume 2.5D floors and offer sim-only code, we employ Splat-Nav [28], a state-of-the-art planner on a pre-built GS map.

Since a pre-built GS map is a mandatory input for Splat-Nav, we provided it with the complete environment map and precise target waypoints. For path planning, Splat-Nav used this complete map, which is ideal conditions for the autonomous baseline, whereas our HIL-GS operator received no prior map and generated its trajectory in an unknown environment. For the reconstruction task, however, both methods started from scratch (zero-base), using only the sensor data gathered along their respective trajectories.

2) Procedures: The task was to reconstruct three target valves on a horizontal pipe (Fig. 4(b)). For each of the five trials, the azimuth of each valve was randomized (-45 or 45 degrees), and both methods were run on the same five randomized layouts to ensure a fair comparison. Evaluation was performed against a ground-truth dataset of 15 images, consisting of five canonical views (front, back, left, right, top) focused on each of the three target valves (Fig. 4(d)).

For Splat-Nav baseline, we set a sequence of six waypoints, placing two on alternating sides of each of the three target valves. This sequence was designed to compel the planner to trace a semicircular path over the pipe, ensuring the camera maintained full coverage of the targets (Fig. 4(c)). However, in pilot tests, Splat-Nav frequently failed to find a feasible path between these consecutive waypoints due to the cluttered environment. We therefore had to manually add intermediate waypoints to guide the planner through difficult sections, providing it with significant human assistance.

For the HIL-GS trials, five non-expert subjects first received a 15-minute familiarization period, then explicitly instructed with the goal: to reconstruct the three target valves safely and quickly

TABLE I
THREE VALVES RECONSTRUCTION TASK COMPARISON: HIL-GS EFFICIENCY METRICS ARE REPORTED AS THE MEAN (STANDARD DEVIATION) ACROSS FIVE NON-EXPERT USERS

Metric	Splat-Nav	HIL-GS
Operation / efficiency		
Total operation time ↓	14m 35s	**10m 48s (2m 32s)**
Movement time ↓	14m 35s	**7m 19s (2m 10s)**
Interaction time ↓	–	3m 29s (30s)
Path planning time ↓	0.174s	–
Reconstruction quality		
PSNR ↑	16.38	**17.70**
SSIM ↑	0.567	**0.602**
LPIPS ↓	0.429	**0.420**

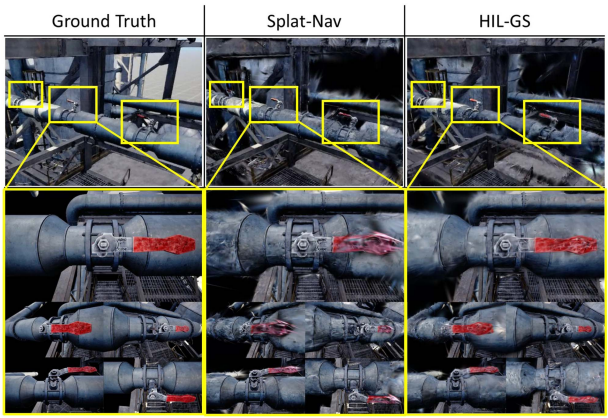

Fig. 5. Comparative result between Splat-Nav & our HIL-GS: Due to its limited trajectory from path planning failures, Splat-Nav baseline (middle) exhibits significant artifacts, whereas HIL-GS (right) achieves a more complete and higher-fidelity result.

within a time limit $T_{max} = 20\ min$. Critically, operators received none of the prior information given to the autonomous baseline (e.g., no pre-built map or target locations), forcing them to rely solely on live reconstruction feedback—a challenging condition designed to mimic a real-world mission in an unknown environment.

3) Results: Fig. 4(c),(e) offer a qualitative comparison of the trajectories. Even with manually-added waypoints, Splat-Nav baseline's path was visibly segmented and composed of discrete point-to-point movements, whereas HIL-GS operators leveraged live feedback to create consistently more fluid and direct trajectories. Table I summarizes the quantitative results. In terms of reconstruction quality, HIL-GS significantly outperformed the baseline across all metrics, achieving a PSNR of 17.70 (+1.32), SSIM of 0.602 (+0.035), and LPIPS of 0.420 (-0.009), which is also shown in Fig. 5. This performance gap stems from the baseline's rigid trajectory, whereas human operators used live feedback to continuously improve the reconstruction with more diverse viewpoints, especially in areas that remained poorly defined. Furthermore, the performance across the five non-expert subjects was highly consistent, as indicated by the low standard deviations in Table I. This consistency, particularly for total operation time (2 m 32 s) and interaction time (30 s), supports the usability and reproducibility of our HIL framework.

Regarding efficiency, HIL-GS was substantially faster, completing the task in 10 m 48 s on average—nearly four minutes shorter than Splat-Nav. This was driven by a remarkable, seven-minute reduction in robot movement time, a direct result of the human operators' ability to plan more efficient, task-focused trajectories. This shorter movement duration is critical as it implies lower energy consumption and more efficient use of onboard computational resources. In terms of safety, HIL-GS operators used the collision-warning module to proactively avoid hazards, while Splat-Nav baseline frequently planned paths into risky areas with map artifacts, triggering the same collision warnings.

Note that Splat-Nav was evaluated under ideal, autonomy-friendly conditions (a pre-built map and a predefined waypoints). That HIL-GS still outperforms the baseline under these conditions strongly demonstrates the practical advantage of human-in-the-loop viewpoint selection for teleoperation in complex environments.

TABLE II
QUANTITATIVE COMPARISON OF RECONSTRUCTION METHODS ACROSS TWO SCENES

Method	Gaussian Splatting SLAM	GS-ICP SLAM	HIL-GS (Ours)
Pumpjack			
PSNR ↑	12.41	13.17	**24.74**
SSIM ↑	0.605	0.639	**0.844**
LPIPS ↓	0.379	0.343	**0.097**
Oil Tank			
PSNR ↑	12.79	14.81	**27.18**
SSIM ↑	0.704	0.749	**0.942**
LPIPS ↓	0.330	0.232	**0.039**

C. Experiment 2

1) Procedures: The second experiment evaluated the accuracy and robustness of our proposed motion-aware sensor fusion algorithm against two vision-only baseline methods (Gaussian Splatting SLAM [22] and GS-ICP SLAM [21]). Reconstruction was performed using the same multi-view images acquired from our HIL framework, ensuring identical viewpoint trajectories for fair comparison. As in the previous experiment, an operator planned and executed safe, task-focused trajectories using the HIL system to densely reconstruct task-critical objects without collisions while minimizing redundant views.

We selected two representative scenes: Pumpjack (large-scale structures) and Oil Tank (rooftop tanks with valves and expansive distant backgrounds). This experiment assessed performance under challenging teleoperation conditions such as aggressive motions (pure rotations, rapid movements), distant backgrounds, and visually complex environments.

2) Results: Fig. 6 and Table II show that our motion-aware sensor fusion method outperformed both Gaussian Splatting SLAM and GS-ICP SLAM baselines across both scenes. Vision-only methods frequently failed during challenging motions (fast

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

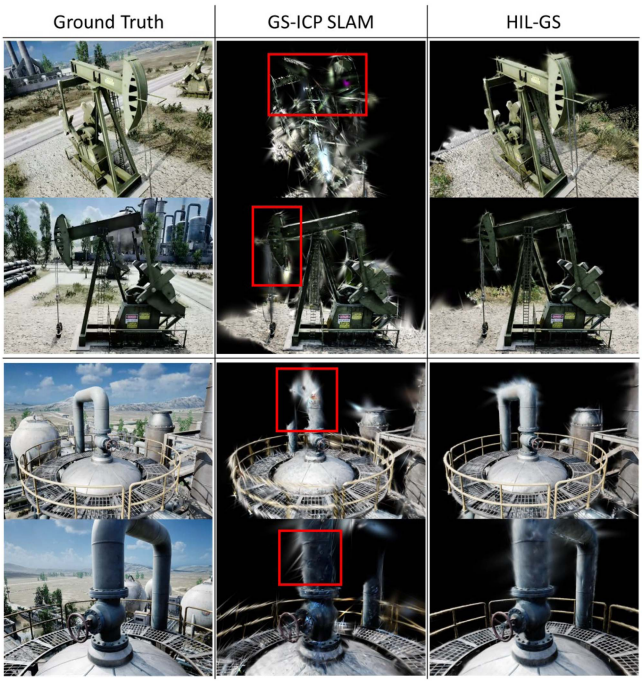

Fig. 6. Reconstruction results across two scenes: Pumpjack (top), and Oil Tank (bottom). Columns show (1) ground-truth camera image, (2) GS-ICP-SLAM, and (3) our motion-aware sensor fusion method. Red boxes indicate areas of distorted, low-fidelity, or even divergent reconstruction. .

linear or rotational-only maneuvers), resulting in incomplete, distorted or even divergent reconstructions. In teleoperation, robots move for manipulation or inspection rather than optimized scanning, leading to abrupt and unstructured trajectories unlike the smooth paths in SLAM benchmarks. Combined with challenging scene conditions — such as distant sparse backgrounds in the Pumpjack and Oil Tank scenes — these factors caused frequent tracking failures, incomplete reconstructions, and severe drift (see supplementary video).

In contrast, our method maintained stable and accurate reconstruction even under these difficult conditions. By integrating proprioceptive motion data, the system prevented tracking failures and reduced drift, enabling reconstruction of both large-scale structures and fine task-critical details with higher fidelity (PSNR, SSIM) and perceptual quality (LPIPS). Quantitatively, our method achieved the highest scores in all metrics (Table II), demonstrating its robustness and suitability for safe, high-fidelity teleoperation mapping in complex industrial environments.

V. Real-World Experiment

A. Experimental Setup

To evaluate the practical feasibility of our reconstruction framework on real robotic teleoperation platforms with actual sensor and motion constraints, we conducted experiments using two distinct robot setups, as illustrated in Fig. 7. The first setup uses a robotic arm (Franka Emika Panda) tasked with reconstructing a remote target object (RC car). The second, more

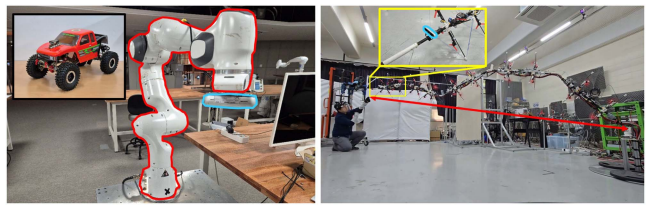

Fig. 7. Real-world experiment setup: The teleoperated robotic arm (Franka Emika Panda) is equipped with an RGB-D camera to reconstruct an RC car (left). 5-meter modular aerial robot is equipped with a front-mounted RGB-D camera to reconstruct the target valve (right). .

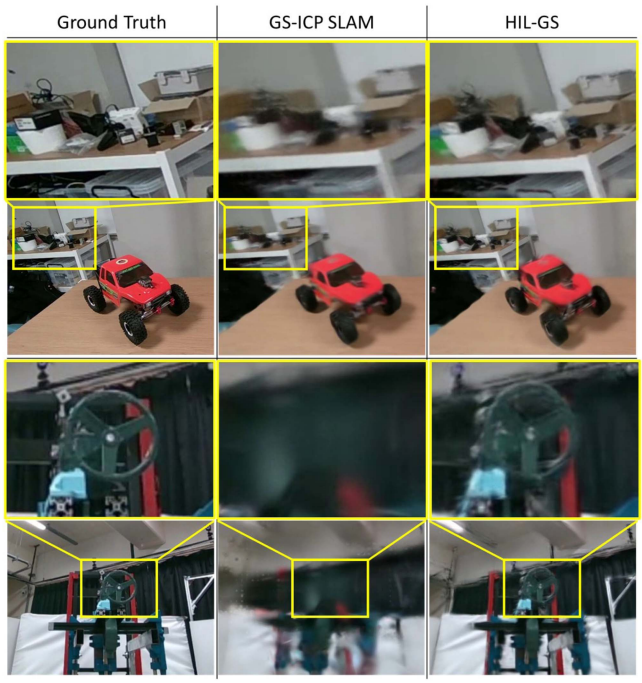

Fig. 8. Real-world reconstruction results for two experimental setups: Teleoperated robotic arm (top) 5-meter modular aerial robot (bottom). Columns show (1) ground-truth camera image, (2) GS-ICP-SLAM, and (3) our motion-aware sensor fusion method. .

complex setup uses LASDRA [29], a 5-meter modular aerial manipulator designed for heavy payloads and omnidirectional control in industrial scenarios; in this experiment, LASDRA was tasked with reconstructing a circular valve to simulate industrial valve-turning tasks. Both platforms were equipped with an onboard RGB-D camera (Intel RealSense D530i), using joint encoders (Franka) or a motion capture system (LASDRA) for proprioceptive data.

B. Results

Fig. 8 presents our real-world reconstruction results. As illustrated, the vision-only baseline exhibited substantial degradation—such as blurred backgrounds and distorted edges—especially under the challenging motions (rapid aerial jitters) and wide depth variations of the LASDRA aerial platform. In contrast, by integrating the robot's motion data, our proposed sensor fusion method produced robust and detailed

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

reconstructions, enabling an accurate representation of intricate geometries critical for remote manipulation tasks.

Quantitatively, our sensor fusion method significantly outperformed the vision-only baseline on both platforms. For the Franka Panda arm, we measured a PSNR of *24.15*, SSIM of *0.835*, and LPIPS of *0.199*. The LASDRA aerial manipulator showed similarly strong results (PSNR *23.27*, SSIM *0.846*, LPIPS *0.149*), highlighting our method's robustness in challenging aerial scenarios.

VI. Conclusion

We introduced the first human-in-the-loop teleoperation framework that integrates motion-aware GS reconstruction with immersive VR-based interfaces and intuitive finger-based controls. Our system enables operators to achieve dense and accurate reconstruction of task-critical objects in challenging industrial environments, validated through both simulation with refinery-scale scenes and real-world experiments on robotic arm and aerial manipulator platforms. Critically, expert-guided planning with predictive and informative displays ensures reconstruction completeness and operational safety by preventing collisions and enabling situationally aware decision-making.

The potential applications of our HIL-GS framework extend to critical domains such as industrial plant inspection, hazardous material handling, or disaster response. This empowers operators to inspect complex geometries from any viewpoint, peer around occlusions, and accurately assess clearances, enabling the safe and efficient completion of intricate tasks that would otherwise be excessively risky.

Future work will extend this framework to real-robot manipulation tasks, integrating dynamic target object segmentation and tracking. We will also explore semi-autonomous functionalities by leveraging our collision-warning feature to generate safe, local collision-avoidance trajectories. This will create a collaborative paradigm where the operator provides high-level goals and the robot handles fine-grained movements, reducing cognitive load and enabling challenging teleoperation tasks.

References

[1] T. Zhou, Q. Zhu, and J. Du, "Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction," *Adv. Eng. Inform.*, vol. 46, 2020, Art. no. 101170.

[2] Y. Su, X. Chen, T. Zhou, C. Pretty, and G. Chase, "Mixed reality-integrated 3D/2D vision mapping for intuitive teleoperation of mobile manipulator," *Robot. Comput.- Integr. Manuf.*, vol. 77, 2022, Art. no. 102332.

[3] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–14, 2023.

[4] V. Patil and M. Hutter, "Radiance fields for robotic teleoperation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 13861–13868.

[5] Y. Li et al., "ActiveSplat: High-fidelity scene reconstruction through active Gaussian splatting," *Robot. Automat. Lett.*, vol. 10, no. 8, pp. 8099–9016, 2025.

[6] B. Axelrod, L. P. Kaelbling, and T. Lozano-Pérez, "Provably safe robot navigation with obstacle uncertainty," *Int. J. Robot. Res.*, vol. 37, no. 13/14, pp. 1760–1774, 2018.

[7] S. Yu, T. Y. Kim, W. W. Park, S. H. Lee, and J. Han, "Intuitive teleoperation with hand-tracking in VR: A study on master–slave system virtualization and 3D workspace visualization," *Int. J. Adv. Manuf. Technol.*, vol. 134, no. 5/6, pp. 2353–2372, 2024.

[8] P. Stotko et al., "A VR system for immersive teleoperation and live exploration with a mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 3630–3637.

[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 *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 99–106.

[10] B. Xie, M. Han, J. Jin, M. Barczyk, and M. Jagersand, "A generative model-based predictive display for robotic teleoperation," in *Proc. Int. Conf. Robot. Automat.*, 2021, pp. 2407–2413.

[11] M. Sakashita, B. T. Kumaravel, N. Marquardt, and A. D. Wilson, "Sharednerf: Leveraging photorealistic and view-dependent rendering for real-time and remote collaboration," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2024, pp. 1–14.

[12] X. Tu et al., "VRSplat: Fast and robust gaussian splatting for virtual reality," in *Proc. ACM Comput. Graph. Interactive Techn.*, vol. 8, no. 1, 2025, pp. 1–22, doi: 10.1145/3728311.

[13] Y. Jiang et al., "VR-GS: A physical dynamics-aware interactive Gaussian splatting system in virtual reality," in *Proc. ACM SIGGRAPH Conf. Papers*, 2024, pp. 1–12.

[14] A. Sidaoui, M. K. Zein, I. H. Elhajj, and D. Asmar, "A-SLAM: Human in-the-loop augmented SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 5245–5251.

[15] S. Nashed and J. Biswas, "Human-in-the-loop SLAM," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 1503–1510.

[16] K. Koide, J. Miura, M. Yokozuka, S. Oishi, and A. Banno, "Interactive 3D graph SLAM for map correction," *Robot. Automat. Lett.*, vol. 6, no. 1, pp. 40–47, 2020.

[17] Z. Ouyang, C. Zhang, and J. Cui, "Semantic SLAM for mobile robot with human-in-the-loop," in *Proc. Int. Conf. Collaborative Comput., Netw., Appl. Worksharing*, 2022, pp. 289–305.

[18] J. Chen, M. Glover, C. Yang, C. Li, Z. Li, and A. Cangelosi, "Development of an immersive interface for robot teleoperation," in *Proc. Towards Auton. Robot. Syst.*, 2017, pp. 1–15.

[19] Y. Li et al., "HSS-SLAM: Human-in-the-loop semantic SLAM represented by superquadrics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 10469–10475.

[20] K. Andersen, S. J. Gaab, J. Sattarvand, and F. C. Harris, "Mets VR: Mining evacuation training simulator in virtual reality for underground mines," in *Proc. Int. Conf. Inf. Technol. New Gener.*, 2020, pp. 325–332.

[21] S. Ha, J. Yeon, and H. Yu, "RGBD GS-ICP SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 180–197.

[22] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 18039–18048.

[23] N. Keetha et al., "SplaTAM: Splat, track & map 3D Gaussians for dense RGB-D SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21357–21366.

[24] Luma AI, "Luma unreal engine plugin v0.41," 2024. Accessed: May 6, 2025. [Online]. Available: https://lumaai.notion.site/Luma-Unreal-Engine-Plugin-0-41-8005919d93444c008982346185e933a1

[25] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "Moveit! Task constructor for task-level motion planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 190–196.

[26] Y. Lee, W. Do, H. Yoon, J. Heo, W. Lee, and D. Lee, "Visual-inertial hand motion tracking with robustness against occlusion, interference, and contact," *Sci. Robot.*, vol. 6, no. 58, 2021, Art. no. eabe1315.

[27] Y. S. Sarkisov et al., "Development of SAM: Cable-suspended aerial manipulator," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 5323–5329.

[28] T. Chen et al., "Splat-NAV: Safe real-time robot navigation in Gaussian splatting maps," *IEEE Trans. Robot.*, vol. 41, pp. 2765–2784, 2025.

[29] H. Yang, S. Park, J. Lee, J. Ahn, D. Son, and D. Lee, "LASDRA: Large-size aerial skeleton system with distributed rotor actuation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7017–7023.