A Unified Framework for Gaussian-Based Scene Representation and Reactive Robot Control

Choi, Ho Jin; Jain, Siddarth

TR2025-096 June 26, 2025

Abstract

Traditional approaches for robotic arm motion planning assume static environments and often depend on prior knowledge about the shape and location of obstacles. Current scene mapping methods enable detailed scene reconstruction, but they are mostly suited for static scenes and they struggle to balance computational efficiency and fidelity. In this paper, we propose a unified framework for Gaussian-based scene rep- resentation and collision-free reactive robot control in unknown environments. We propose a real-time method for dynamic scene reconstruction from RGB-D images, enhancing the 3D Gaussian Splatting with key improvements under a fixed budget of Gaussians. Additionally, we introduce a technique for computing the signed distance function of the reconstructed environment using isotropic Gaussians, providing reduced computational complexity and smooth interpolation for computations of collision probability and reactive control. Our method demonstrates promising results in robot experiments across a range of environments, both in simulations and real-world settings.

Robotics: Science and Systems (RSS) 2025 Workshop on Gaussian Representations for Robot Autonomy

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

A Unified Framework for Gaussian-Based Scene Representation and Reactive Robot Control

Ho Jin Choi
University of Pennsylvania
Philadelphia, Pennsylvania, USA
Email: cr139139@seas.upenn.edu

Siddarth Jain
Mitsubishi Electric Research Labs (MERL)
Cambridge, Massachusetts, USA
Email: sjain@merl.com

***Abstract*—Traditional approaches for robotic arm motion planning assume static environments and often depend on prior knowledge about the shape and location of obstacles. Current scene mapping methods enable detailed scene reconstruction, but they are mostly suited for static scenes and they struggle to balance computational efficiency and fidelity. In this paper, we propose a unified framework for Gaussian-based scene representation and collision-free reactive robot control in unknown environments. We propose a real-time method for dynamic scene reconstruction from RGB-D images, enhancing the 3D Gaussian Splatting with key improvements under a fixed budget of Gaussians. Additionally, we introduce a technique for computing the signed distance function of the reconstructed environment using isotropic Gaussians, providing reduced computational complexity and smooth interpolation for computations of collision probability and reactive control. Our method demonstrates promising results in robot experiments across a range of environments, both in simulations and real-world settings.**

I. Introduction

Generating collision-free motions for a robotic manipulator in unknown environments remains a significant challenge. While robots perform reliably in structured settings like factory floors—where tasks are repetitive and object locations are fixed—these environments are often engineered at high cost to ensure repeatability. In contrast, real-world environments are unstructured, dynamic, and lack prior information, requiring robots to jointly perceive the environment and adapt to changes in real time. It is challenging for any single approach to satisfy multiple considerations and existing approaches rely on strong assumptions about obstacle representation and collision checking, limiting their ability to operate directly in novel scenes using raw sensor observations.

Perception for scene reconstruction in robotics has traditionally been built on point-cloud and voxel-based representations [1, 2]. Point cloud approaches, while being flexible, lack inherent surface representations and can struggle with occlusions and temporal consistency. On the other hand, voxel-based methods struggle to balance voxel granularity with computational efficiency. Recent breakthroughs—notably Neural Radiance Fields (NeRF) [3] and 3D Gaussian Splatting (3D-GS) [4] have demonstrated remarkable capabilities in capturing high-fidelity scene representations. NeRF excels at modeling complex environments through implicit representations, while 3D-GS offers an explicit and optimizable framework. Despite their potential, applying these methods directly to robotics remains challenging due to several limitations: high computational demands, assumptions of static environments, and insufficient integration with robotic frameworks.

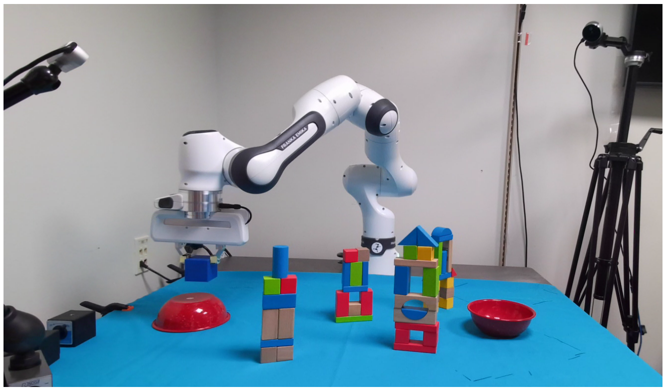


Fig. 1: Robot setup in an unstructured environment with a variety of obstacles. The goal is to reconstruct the scene using RGB-D cameras and ensure collision-free motion control for effective task execution.

This paper introduces a unified framework that combines real-time scene reconstruction using Gaussian representations with reactive robotic control to perform collision-free robot motions in unknown environments. Our framework leverages the strengths of 3D Gaussian Splatting while addressing its limitations for robotic applications through several key contributions. Our first contribution extends the traditional 3D-GS with a novel voxel-based filtering and dynamic Gaussian relocation strategy. This hybrid approach enables fast and efficient creation and updating of environmental changes from RGB-D image streams while maintaining the fidelity of reconstruction. This contribution can handle gradual changes and introductions of new entities in the environment (e.g., obstacles or a human). Next, we introduce a new method for computing continuous signed distance functions from isotropic Gaussians, providing stable and differentiable collision probability estimates. This contribution bridges the gap between traditional discrete distance fields and modern learning-based representations, offering computational efficiency and numerical stability. The continuous nature of the proposed representation enables smooth gradient information for robotic control

This research was completed during Ho Jin Choi's internship at MERL.

while maintaining accuracy in distance estimates. We combine the continuous distance representations with control barrier functions in a unified framework for perception and control. Our experimental results, both in simulation and on a physical robot (Figure 1), are promising. Notably, they represent the first evaluation of a system capable of handling both translation and rotation for collision-free robotic control in unknown environments, while reconstructing scenes in real-time from an input stream of RGB-D images.

The rest of the article is structured as follows: Section II reviews related work. Section III introduces the problem statement and outlines the proposed method. Section IV details the experiments and presents the results, which are further discussed in Section V with limitations. Finally, Section VI concludes the paper and suggests directions for future research.

II. Related work

Scene reconstruction plays a crucial role in robotic perception, allowing robots to comprehend and interact with their environment. Data structures such as point clouds [5], voxel grids [6], meshes [7], and Signed Distance Functions [8] have been used to model robot environments. Octomap [2], which divides the space into grids, remains widely used for scene representation and collision detection [9].

Recent advancements in differentiable rendering have significantly enhanced high-fidelity representations. Neural Radiance Fields (NeRF) [3], model environments as continuous functions, parameterized by neural networks to capture complex geometries and fine details. This approach has found applications in robotics, including pose estimation [10], SLAM [11], and representation learning [12]. While NeRF excels at generating photorealistic renderings, there is an increasing need for faster and more efficient methods for latency sensitive applications. 3D Gaussian Splatting (3D-GS) [4] emerged to address this challenge by introducing an advanced explicit scene representation.

Recent studies have extended Gaussian Splatting (3D-GS) to robotics. ManiGaussian [13] introduces a framework that parameterize implicit Gaussian points to predict future states and actions. Splat-MOVER [14] builds a modular stack for open-vocabulary manipulation. Robo-GS [15] combines Gaussians, grids, and pixels to reconstruct manipulable robot arms, enabling high-fidelity Real2Sim transfer. SplatSim [16] jointly reconstructs robots and objects for Sim2Real tasks. However, existing methods often require dense multi-view inputs or costly scene-specific optimization, making them impractical for real-world, dynamic, or unstructured environments. Moreover, inconsistent rendering and free-space artifacts degrade spatial accuracy—an issue critical in robotics. Our approach addresses these limitations through voxel-based filtering, dynamic Gaussian relocation, and a reformulation of isotropic Gaussians into a unified distance and collision probability field, enhancing robustness and real-world applicability.

III. Method

Given RGB-D image streams from multiple cameras our method automatically represent scenes using 3D Gaussians within a unified framework. Our framework integrates real-time scene reconstruction with collision-free reactive robotic control or motion planning (Figure 2). We concentrate on the robot's workspace, which encompasses the area where the robot can interact with objects. In the following sections, we introduce our proposed framework to tackle the challenges associated with using Gaussian representations in robotics.

A. Problem Statement

We addresses the problem of maintaining an efficient scene representation of a continuously changing workspace $\mathcal{W} \subset \mathbb{R}^3$ for collision-free robotic control in the workspace of the robot. We approximate $\mathcal{W}$ using a set of isotropic Gaussians $\mathcal{G} = \{G_i = (p_i, r_i, ...)\}_{i=1}^N$, where $p_i \in \mathbb{R}^3$ and $r_i > 0$ represent the center and radius of each Gaussian, respectively. The key challenge is to construct and efficiently update a proxy Signed Distance Function $\mathrm{SDF}(x) : \mathcal{W} \rightarrow \mathbb{R}$ that adapts to changes in $\mathcal{G}$ using Gaussian Process Distance Fields. This $\mathrm{SDF}(x)$ needs to balance accuracy and computational efficiency. The next goal is to enable collision-free, real-time robot control by using the distance representation and its gradient to formulate a reactive collision-avoidance constraint.

B. 3D Gaussian Scene Representation

We represent the scene using isotropic 3D Gaussians [17, 18] due to their computational simplicity and efficiency in rendering and distance calculations. Each Gaussian, G_i, is defined by 8 parameters: position p_i (3), RGB color c_i (3), radius r_i (1), and opacity α_i (1). To simplify calculations, the covariance matrix $\Sigma_i = r_i^2 I$, where I is the 3x3 identity matrix, assumes spherical symmetry. In later stages, we add additional parameters: density ρ_i and segmentation label s_i.

This Gaussian representation enables differentiable rendering of color and depth from any viewpoint using RGB-D streams, allowing scene optimization with observed images and poses. In camera coordinates, the Gaussians have covariance Σ':

$$\Sigma' = JW\Sigma W^T J^T \tag{1}$$

where J is the Jacobian of the projective transformation, and W is the view transformation matrix. The pixel color is computed by alpha-blending Gaussians in front-to-back order:

$$\hat{C} = \Sigma_i c_i \alpha_i \Pi_{j=1}^{i-1}(1 - \alpha_i) \tag{2}$$

Depth is rendered using the center depth d_i of each Gaussian:

$$\hat{D} = \Sigma_i d_i \alpha_i \Pi_{j=1}^{i-1}(1 - \alpha_i) \tag{3}$$

C. Gaussian Splatting for Robotics

Multiple cameras and randomly initialized Gaussians enable the original GS to reconstruct a scene, but dynamic environments with changing objects require continuous updates.

We initialize the Gaussians based on the point cloud data from RGBD cameras, with initial opacity of $\alpha_{\text{init}} = 0.5$ as

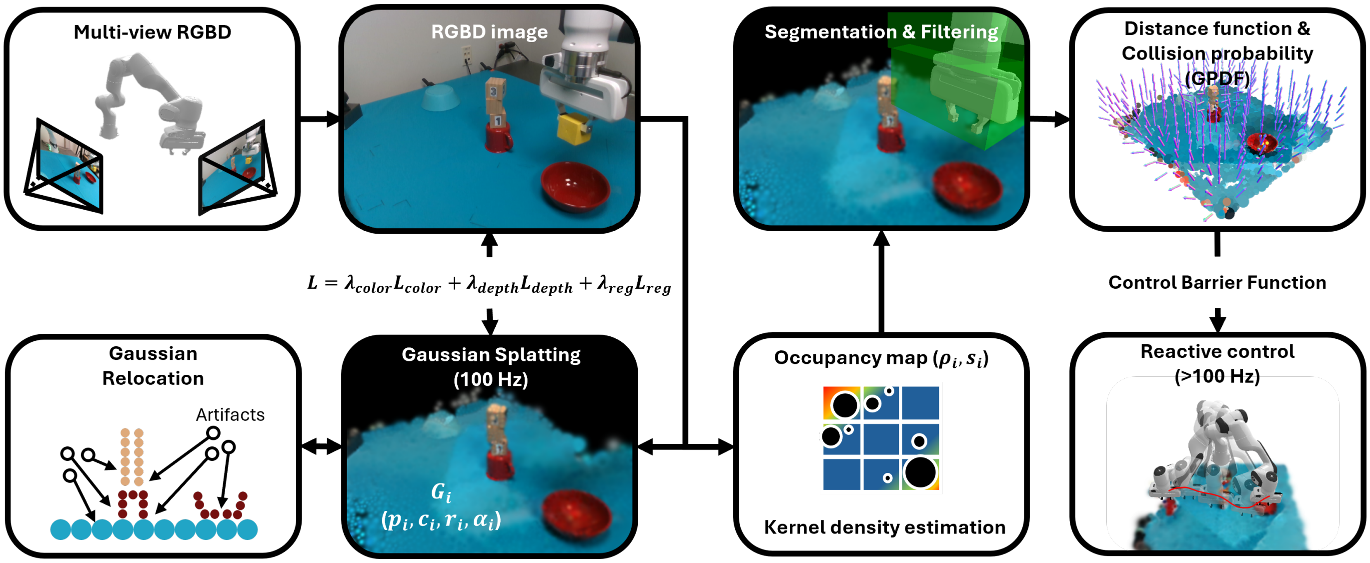

Multi-view RGBD
RGBD image
Segmentation & Filtering
Distance function & Collision probability (GPDF)
$L = \lambda_{color} L_{color} + \lambda_{depth} L_{depth} + \lambda_{reg} L_{reg}$
Control Barrier Function
Gaussian Relocation
Artifacts
Gaussian Splatting (100 Hz)
G_i
$(p_i, c_i, r_i, \alpha_i)$
Occupancy map (ρ_i, s_i)
Kernel density estimation
Reactive control (>100 Hz)

Fig. 2: An overview of our proposed framework, including scene reconstruction via isotropic Gaussian splatting, followed by occupancy-based density estimation and segmentation, and integration with a Gaussian Process Distance Function to generate a continuous control barrier function.

in [17]. The radius is set based on the pixel size, clamped to a maximum value $r = \min(D_{\text{gt}}/f, r_{\max})$, where f is the camera's focal length. Gaussian parameters are optimized by rendering images and comparing them to the ground truth [4]. The loss function used for optimization is:

$$\begin{aligned} L_{\text{color}} &= (1-\lambda)L_1(\hat{C}, C_{\text{gt}}) + \lambda L_{\text{D-SSIM}}(\hat{C}, C_{\text{gt}}) \\ L_{\text{depth}} &= L_1(\hat{D}, D_{\text{gt}}) \\ L_{\text{reg}} &= \Sigma_i |\alpha_i| \\ L &= \lambda_{\text{color}} L_{\text{color}} + \lambda_{\text{depth}} L_{\text{depth}} + \lambda_{\text{reg}} L_{\text{reg}} \end{aligned} \tag{4}$$

where L_1 is least absolute deviations and $L_{\text{D-SSIM}}$ is a loss for difference of structural similarity. Here, $\lambda = 0.2$, $\lambda_{\text{color}} = \lambda_{\text{depth}} = 1$, and $\lambda_{\text{reg}} = 0.02$. We use RGBD images to reduce the reconstruction error for high fidelity. The regularization term, L_{reg}, encourages fewer Gaussians. Noise is added after each Adam optimizer update. In dynamic situations, Gaussians need to be added, removed, or relocated beyond the optimization. A mask is first generated to identify where to add new Gaussians [17]:

$$M_{\text{add}} = (D_{\text{gt}} < \hat{D}) \wedge (L_1(\hat{D}, D_{\text{gt}}) > \lambda_{\text{MDE}}\text{MDE}) \tag{5}$$

This mask adds Gaussians where the ground truth depth is closer than the rendered depth, and the depth error exceeds λ_{MDE} times the median depth error (MDE). Gaussian Splatting generates artifacts that can potentially cause false positives collision. To mitigate this issue, we apply a minimum opacity threshold of $\alpha_{\min} = 0.1$ to filter out transparent Gaussians. Additionally, we use a GPU-based occupancy grid with voxel sizes at least as large as the maximum Gaussian radius. This enables us to evaluate the occupancy values (ranging from 0 to 1) of neighboring voxels and remove Gaussians located in free (< 0.3) or uncertain (< 0.7) regions. Gaussians outside the robot's reachable workspace are also removed. We relocate Gaussians to adapt to newly appearing objects using the mask M_{add} and RGB-D image. To maintain computational efficiency, we impose a limit on the maximum number of Gaussians. If the total Gaussian count is below the threshold, new Gaussians are added and dead ones repositioned. Once the limit is reached, only dead Gaussians are replaced, maintaining a fixed count while optimizing distribution.

To enable obstacle avoidance and interaction, the robot must identify Gaussians representing itself. Segmenting from 2D images misses internal Gaussians, causing false collisions. Instead of complex learning-based methods, we use an occupancy grid and the robot's AABB: voxels intersecting the AABB are found, followed by a fixed-radius search for intersecting Gaussians. We use 2-bit encoding to categorize different entities: objects are 00, the robot is 01. Voxel grids are updated using atomic OR operations to track which entity occupies each voxel. As the robot moves quickly, Gaussians may leave "trails" where they were previously, which aren't updated quickly enough by the Gaussian Splatting for a proper tracking. To address this, we add a removal mask for Gaussians previously classified as robot that have become objects.

D. SDF and Collision Probability

SDF provides information about whether a point is inside, outside, or on the surface of an object, enabling more nuanced robotic interactions. Isotropic Gaussians can define a surface boundary by their radii r_i, naturally producing an SDF [19]. The SDF at a query point x in 3D space is:

$$\mathrm{SDF(x)} = \min_{\mathrm{i}}(\|\mathrm{x} - \mathrm{p_i}\|_2 - \mathrm{r_i}) \tag{6}$$

Here, p_i and r_i denote the center and radius of the i-th Gaussian. This method suffers from discontinuities, nondifferentiability, and poor surface interpolation under sparse Gaussians. To address these issues, we extend the Gaussian process distance field (GPDF) [20] from point cloud data to include radii. We use it as a proxy for SDF and we improve its numerical stability and computational efficiency by modeling the occupancy field $o(x)$ as a Gaussian process:

$$o(x) \sim \mathcal{GP}(0, \mathbf{k}(x, p)) \tag{7}$$

Here, $x \in \mathbb{R}^3$ is the query point, and $p \in \mathbb{R}^3$ is the center of a Gaussian sphere, and the kernel function $\mathbf{k}(x, p) = \mathbf{k}_d(\mathbf{d}(x, p)) = \exp(-\mathbf{d}(x, p)/l)$ is a Matérn kernel with interpolation parameter $l > 0$ where $\mathbf{d}(x, p) = \|x - p\|_2$. The inferred occupancy $\hat{o}(x)$ is computed as:

$$\hat{o}(x) = \mathbf{k}(x, \mathbf{P})(\mathbf{K}(\mathbf{P}, \mathbf{P}) + \epsilon I)^{-1}\mathbf{y} > 0 \tag{8}$$

where $\mathbf{P} = [p_1, p_2, \ldots, p_n]^T$ are the sphere centers, $\mathbf{y} = [\mathbf{k}_d(-r_1), \mathbf{k}_d(-r_2), \ldots, \mathbf{k}_d(-r_n)]^T$ are the kernel evaluations at negative radii, $\epsilon > 0$ is a small noise for observation. The occupancy describes the point's location relative to the object:

$$\hat{o}(x) < 1 : \text{outside}, \quad = 1 : \text{surface}, \quad > 1 : \text{inside}$$

Although a Gaussian Process (GP) output may not always be positive despite positive observations $\mathbf{y}$, $\hat{o}(x)$ generally stays positive. This positivity is guaranteed under an approximation introduced later. The SDF is defined as:

$$\widehat{\text{SDF}}(x) = \mathbf{r}(\hat{o}(x)) \tag{9}$$

where $\mathbf{r}$ is the inverse function of $\mathbf{k}_d$. The gradient of SDF is

$$\begin{aligned} \nabla\widehat{\text{SDF}}(x) &= \frac{\partial r}{\partial \hat{o}} \nabla \hat{o}(x) \\ \nabla \hat{o}(x) &= \nabla \mathbf{k}(x, \mathbf{P})(\mathbf{K}(\mathbf{P}, \mathbf{P}) + \epsilon I)^{-1}\mathbf{y} \end{aligned} \tag{10}$$

To satisfy the Eikonal equation, the gradient is normalized:

$$\nabla\widehat{\text{SDF}}(x) = \frac{\nabla \hat{o}(x)}{\|\nabla \hat{o}(x)\|_2} \tag{11}$$

The interpolated proxy SDF typically underestimates true surface distance. While sufficient for barrier functions, sphere marching refines the SDF in 3–5 iterations for higher accuracy [21]. Alternatively, kernels such as the Gaussian $\mathbf{k}_d(\mathbf{d}) = \exp(-\mathbf{d}^2/(2l^2))$ yield sharper distance fields via a reverting function but may cause numerical discontinuities.

1) Numerical stability: GPDF faces numerical instability from $\mathbf{r}(x) = -l\log(x)$, which diverges for points far from P [22]. We mitigate this using the following trick:

$$\begin{aligned} &\mathbf{M} = [m_1, m_2, \ldots, m_n]^T = (\mathbf{K}(\mathbf{P}, \mathbf{P}) + \epsilon I)^{-1}\mathbf{y} \\ &\hat{o}(x) = \mathbf{k}(x, p_1)m_1 + \mathbf{k}(x, p_2)m_2 + \cdots + \mathbf{k}(x, p_n)m_n \\ &\widehat{\text{SDF}}(x) = -l\log(\mathbf{k}(x, p_1)m_1 + \cdots + \mathbf{k}(x, p_n)m_n) \\ &\;= \mathbf{d}(x, p_1) - l\log(m_1 + \cdots + \mathbf{k}_d(\mathbf{d}(x, p_n) - \mathbf{d}(x, p_1))m_n) \end{aligned} \tag{12}$$

where $\mathbf{d}(x, p_1)$ is the minimum distance between x and all points in $\mathbf{P}$. Although the log argument may not always be positive since $\mathbf{M}$ can contain negative elements, it usually is, and suitable approximations (discussed later) ensure positivity.

2) Collision probability: The variance of the Gaussian process provides a measure of collision probability:

$$\mathbf{var}(o(x)) = \mathbf{k}(x, x) - \mathbf{k}(x, \mathbf{P})(\mathbf{K}(\mathbf{P}, \mathbf{P}) + \epsilon I)^{-1}\mathbf{k}(\mathbf{P}, x) \tag{13}$$

The collision probability $P(o(x) \geq 1)$ where $o(x) > 0$ is:

$$P(o(x) \geq 1) = \frac{1 - \Phi\left(\frac{1 - \hat{o}(x)}{\sqrt{\mathbf{var}(o(x))}}\right)}{1 - \Phi\left(\frac{0 - \hat{o}(x)}{\sqrt{\mathbf{var}(o(x))}}\right)} \tag{14}$$

Here, Φ represents the cumulative distribution function of the standard normal distribution. However, when x is far from $\mathbf{P}$, $\hat{o}(x)$ and $\mathbf{var}(o(x))$ approach 0 and 1, respectively, causing $P(o(x) \geq 1)$ to have a non-zero probability of approximately $(1 - \Phi(1))/(1 - \Phi(0)) \approx 0.3174$. This is because $\mathbf{var}(o(x))$ does not account for the scale of y. To ensure the probability decreases to near zero as x moves away from $\mathbf{P}$, we divide the variance by a constant factor (e.g., we use 9).

3) Approximation: To reduce the complexity of matrix inversion in Gaussian processes, we apply a mass-lumped matrix approximation [23]. We approximate the kernel matrix $\mathcal{K}(\mathcal{P}, \mathcal{P})$ by its row sums, forming a diagonal matrix $\mathcal{D}(\mathcal{P}, \mathcal{P})$:

$$\begin{aligned} &\mathbf{K}(\mathbf{P}, \mathbf{P}) \approx \mathbf{D}(\mathbf{P}, \mathbf{P}) \\ &\quad = \mathrm{diag}\left(\left[\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathbf{k}(\mathrm{p}_1, \mathrm{p}_\mathrm{i}), \ldots, \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathbf{k}(\mathrm{p}_\mathrm{n}, \mathrm{p}_\mathrm{i})\right]^{\mathrm{T}}\right) \end{aligned} \tag{15}$$

This approximation performs kernel regression on the Gaussian process implicit features $\mathbf{F}$, producing new features $\mathbf{F}'$.

$$\begin{aligned} &\mathbf{K}(\mathbf{P}, \mathbf{P})\mathbf{F} = \mathbf{y}, \quad \mathbf{D}(\mathbf{P}, \mathbf{P})\mathbf{F}' = \mathbf{y} \\ &\Rightarrow \mathbf{F}' = \mathbf{D}^{-1}\mathbf{K}\mathbf{F} = \begin{bmatrix} \frac{\mathbf{k}(p_1, p_1)}{\sum_{i=1}^n \mathbf{k}(p_1, p_i)} & \cdots & \frac{\mathbf{k}(p_1, p_n)}{\sum_{i=1}^n \mathbf{k}(p_1, p_i)} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{k}(p_n, p_1)}{\sum_{i=1}^n \mathbf{k}(p_n, p_i)} & \cdots & \frac{\mathbf{k}(p_n, p_n)}{\sum_{i=1}^n \mathbf{k}(p_n, p_i)} \end{bmatrix} \mathbf{F} \end{aligned} \tag{16}$$

This reduces model complexity to $O(n^2)$ and ensures $\hat{o} > 0$. The diagonal terms approximate kernel density, e.g., $\sum_{i=1}^n \mathbf{k}(p_1, p_i)$ estimates the density at p_1. To simplify further, we approximate density via a uniform voxel grid: rasterize $\mathbf{P}$ using trilinear interpolation, apply a GPDF-based kernel convolution, then interpolate back to get densities ρ_i. This reduces complexity to linear in both the number of points and voxels.

E. Collision Avoidance with Reactive Control

We use SDF as a control barrier function for a Quadratic Programming Inverse Kinematics (QP-IK) controller [24, 25]. This method enables reactive and collision-free control in dynamic environments. The QP-IK controller is:

$$\begin{aligned}
\min_{\dot q,\dot{\mathbf{x}}}\quad & (\dot{\mathbf{x}}_{\text{ref}}-\dot{\mathbf{x}})^T Q(\dot{\mathbf{x}}_{\text{ref}}-\dot{\mathbf{x}}) + \delta_{\mathbf{x}}^T R\delta_{\mathbf{x}} + \delta_{\text{ext}}^T S\delta_{\text{ext}} \\
\text{s.t.}\quad & \dot{\mathbf{x}} = J(q)\dot q + \delta_{\mathbf{x}} \qquad \text{(Forward kinematics)} \\
& -\dot q \geq -(q_{\max}-q), \quad \text{(Joint upper limit)} \\
& \dot q \geq -(q-q_{\min}), \quad \text{(Joint lower limit)} \\
& -\dot q_{\max} \leq \dot q \leq \dot q_{\max}, \quad \text{(Joint velocity limit)} \\
& \nabla_q d_{\text{ext}}(q)\dot q \geq -(d_{\text{ext}}(q)-\epsilon)+\delta_{\text{ext}}, \\
& \text{(External collision constraint)} \\
& \nabla_q d_{\text{self}}(q)\dot q \geq -(d_{\text{self}}(q)-\epsilon) \\
& \text{(Self collision constraint)}
\end{aligned} \tag{17}$$

Here, $\dot{\mathbf{x}}_{\text{ref}} \in \mathbb{R}^6$ is the desired end-effector velocity, $J(q) \in \mathbb{R}^{6\times 7}$ the Jacobian, $\delta_{\mathbf{x}} \in \mathbb{R}^6$ and $\delta_{\text{ext}} \in \mathbb{R}^7$ are slack variables, and $Q, R, S \succ 0$ are weights. Joint limits are $q_{\min}, q_{\max}$, and $\dot q_{\max}$. Collision distances and gradients are $d_{\text{ext}}(q), \nabla_q d_{\text{ext}}(q)$ and $d_{\text{self}}(q), \nabla_q d_{\text{self}}(q)$. The cost includes tracking error, slack penalties for kinematics, and soft collision avoidance, with optional terms for velocity smoothness and home pose bias. Collisions are modeled using spheres [26], and QP solves at > 100 Hz using CVXPY [27] with state and Gaussian updates.

IV. Experiments & Results

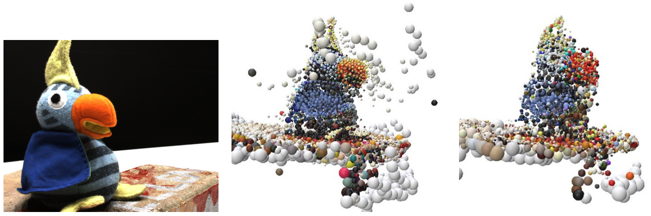
We evaluate our framework on scene reconstruction quality and collision-free motion generation in unknown environments, using both simulation [28] and real-world tests with diverse obstacles. Our experiments use a 7-DOF Franka Emika Panda robot and Intel RealSense D435 RGB-D cameras.

A. Scene Reconstruction Quality

We evaluate scene reconstruction quality using the DTU MVS dataset [29]. Since DTU lacks native depth images, we use depth maps from [30] to synthesize RGB-D data. We benchmark our method against 3D-GS [4] for scene reconstruction quality. Both methods are initialized with 10,000 Gaussians and fitted using identical loss functions. While 3D-GS dynamically increases its count of Gaussians (50,000–100,000), our method maintain a fixed budget of Gaussians, making it more suitable for real-time applications. We chose widely used peak signal-to-noise ratio (PSNR) metric to validate the scene reconstruction quality. Table I reports the average Peak Signal-to-Noise Ratio (PSNR). Our method achieves competitive PSNR. Figure 3 illustrates qualitative comparison. Note that while scene reconstruction quality focuses on the visual fidelity, for robotics, evaluating interaction with the environment is essential, especially collision avoidance for safe and efficient operation.

TABLE I: Quantitative results on the DTU MVS dataset [29].

Method	PSNR ↑
3D-GS	23.27
Ours	23.33



(a) Dataset image (b) 3D-GS [4] (c) Ours

Fig. 3: The qualitative comparison of our method with 3D-GS on the DTU MVS dataset [29].

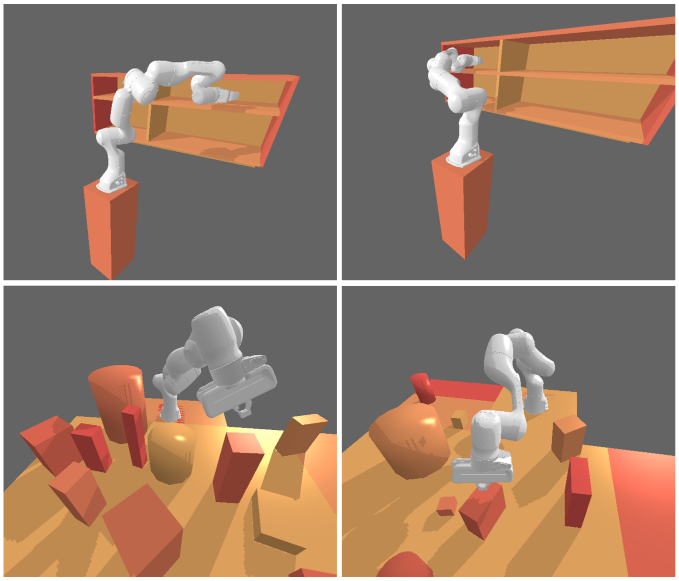


Fig. 4: Samples of Cubby and Tabletop test environments [28].

B. Simulation Robot Experiments

Reconstruction quality mainly measures fidelity but not its usefulness for robotics. We tested our framework to examine the utility of the reconstructed scene for robotic tasks in two simulation environments [28], *Cubby* and *Tabletop* scene. Our framework was tested in 942 trials on these unknown environments (Figure 4), given only start/goal configurations and RGB-D streams. Our framework, combined with a BiRRT motion planner using the Gaussian Process Distance Field (GPDF) for collision checking (GPDF $> 0 =$ valid) with 2cm voxels, was benchmarked against 3D-GS [4]. For this experiment, the reconstruction was performed first, followed by motion planning on the resulting scene. Success requires reaching the target pose within 1° rotation and 3 cm translation without collisions. We varied the number of RGB-D cameras (2–6) for reconstruction.

Table II shows success rates across different settings. Our method outperforms the baseline, requiring fewer cameras and showing greater robustness and efficiency. 3D-GS introduces artifacts causing false collisions, particularly at the start and end-effector positions, while our approach yields more reliable representations. Additionally, our method remains faster, sustaining approximately 240 Hz for single-view iterations, with the limited budget of Gaussians.

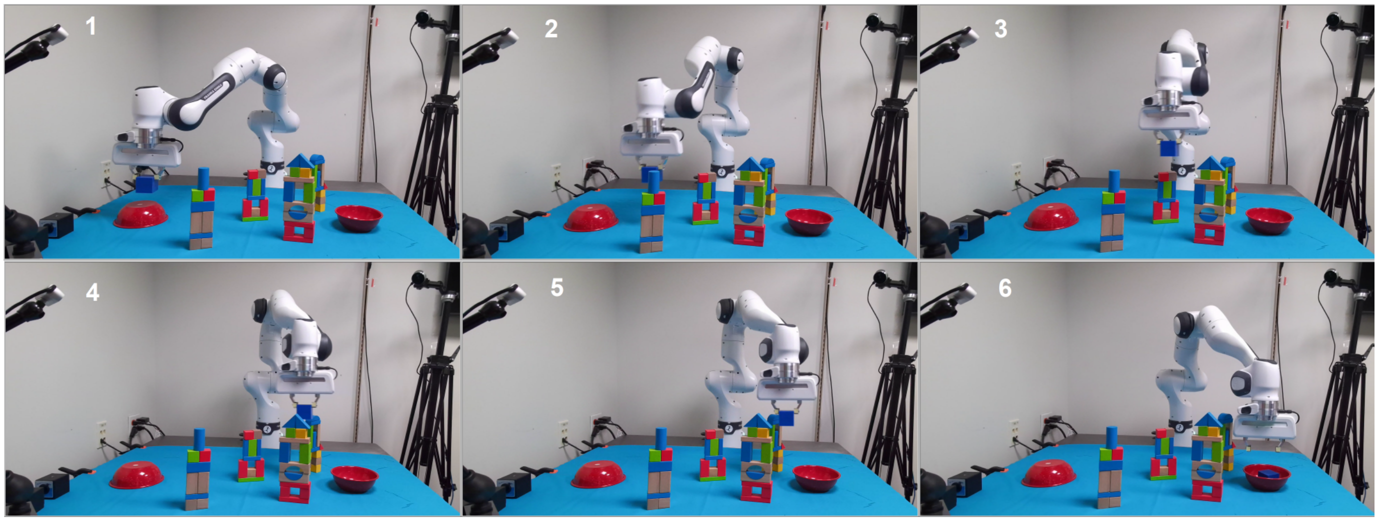


Fig. 5: Qualitative results on real world experiment featuring an unknown environment with a diverse set of obstacles. The figure shows a sampling of keyframes (1-6) selected from the robot motion sequence to illustrate the task progression.

TABLE II: Success rates (%) with motion planning across simulated environments.

Number of Cameras	2	4	6
Tabletop Environment (546 trials)			
3D-GS w/ MP (%)	50.59	78.43	33.88
Ours w/ MP (%)	**95.28**	**92.87**	**98.17**
Cubby Environment (174 trials)			
3D-GS w/ MP (%)	6.45	1.24	0.00
Ours w/ MP (%)	**95.48**	**87.58**	**88.51**
Merged Cubby Environment (222 trials)			
3D-GS w/ MP (%)	8.04	4.19	0.45
Ours w/ MP (%)	**93.97**	**94.42**	**96.85**
Average Across Environments (942 trials)			
3D-GS w/ MP (%)	32.83	48.13	19.75
Ours w/ MP (%)	**95.01**	**92.32**	**96.07**

C. Real Robot Experiments

In addition to evaluating our framework in simulation environments, we conducted real-world experiments to validate its effectiveness for reactive control in unknown environment. We tested our framework with reactive control strategy (QP-IK) in a pick-and-place task on an environment containing a diverse set of obstacles (Figure 5). The task of the robot is to transport the blue cube into the bowl on the right—without collision. The scene is reconstructed on the fly and used directly by the controller, unlike the sampling-based motion planners in the previous experiments. The robot had no prior knowledge of the obstacles' locations or shapes. For scene reconstruction, we employed three Intel RealSense D435 RGB-D cameras with our framework. In our tests we conducted four trials, and we configured the system with a maximum of 5,000 Gaussians and set the Gaussian Process interpolation length scale to 2 cm. Figure 5 illustrates qualitative results with our framework showing motion execution keyframes for the task. Despite the environment being unknown—the robot was able to successfully avoid obstacles and complete the task for all trials. The success of the approach is highly dependent on the quality of the scene reconstruction, as it operate under strict collision constraints.

V. Discussion

While effective, our framework has certain limitations—namely, hyperparameter tuning and the simplification of Gaussian distributions as ellipsoids, which overlooks their full probabilistic characteristics. In future work, we aim to compute distances between Gaussians without relying on point-radius simplifications, thereby preserving richer probabilistic information and increasing the reconstruction accuracy. We also plan to explore image-level spatial filtering to simplify the pipeline and eliminate the need for memory-consuming voxel grids. Additionally, we will evaluate our framework in dynamic environments, such as human-robot collaborative scenarios, to demonstrate its capability to adapt to dynamic changes in the environment.

VI. Conclusion

We have presented a novel unified framework for real-time Gaussian-Based scene reconstruction and reactive Control for robotics. Our key contributions include a memory-efficient Gaussian update strategy, a numerically stable and continuous SDF formulation based on Gaussian Process Distance Fields (GPDF), and a reactive controller that can ensure collision-free reactive motion in unknown environments. Unlike prior approaches, our method operates directly from RGB-D inputs and supports both motion planning and reactive control in unstructured, previously unseen environments. Future work will focus on improved uncertainty modeling, and evaluation on dynamic settings such as human-robot collaborative scenarios.

References

[1] Matteo Palieri, Benjamin Morrell, Abhishek Thakur, Kamak Ebadi, Jeremy Nash, Arghya Chatterjee, Christoforos Kanellakis, Luca Carlone, Cataldo Guaragnella, and Ali-akbar Agha-Mohammadi. Locus: A multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time. *IEEE Robotics and Automation Letters*, 6(2):421–428, 2020.

[2] Armin Hornung, Kai M. Wurm, Maren Bennewitz, C. Stachniss, and Wolfram Burgard. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34:189 – 206, 2013.

[3] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65:99 – 106, 2020.

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023.

[5] Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. *Advances in Neural Information Processing Systems*, 37:77799–77830, 2024.

[6] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[7] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018.

[8] Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan Nieto. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. In *Conference on Robot Learning*, pages 1602–1611. PMLR, 2021.

[9] D.M. Coleman, Ioan Alexandru Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *ArXiv*, abs/1404.3785, 2014.

[10] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. in 2021 ieee. In *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330.

[11] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6229–6238, 2021.

[12] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. Nerf-supervision: Learning dense object descriptors from neural radiance fields. In *2022 international conference on robotics and automation (ICRA)*, pages 6496–6503. IEEE, 2022.

[13] Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *European Conference on Computer Vision*, pages 349–366. Springer, 2024.

[14] Ola Shorinwa, Johnathan Tucker, Aliyah Smith, Aiden Swann, Timothy Chen, Roya Firoozi, Monroe Kennedy III, and Mac Schwager. Splat-mover: Multi-stage, open-vocabulary robotic manipulation via editable gaussian splatting. *arXiv preprint arXiv:2405.04378*, 2024.

[15] Haozhe Lou, Yurong Liu, Yike Pan, Yiran Geng, Jianteng Chen, Wenlong Ma, Chenglong Li, Lin Wang, Hengzhen Feng, Lu Shi, et al. Robo-gs: A physics consistent spatial-temporal model for robotic arm with hybrid representation. *arXiv preprint arXiv:2408.14873*, 2024.

[16] Mohammad Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhisesh Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting. *arXiv preprint arXiv:2409.10161*, 2024.

[17] Nikhil Varma Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. *ArXiv*, abs/2312.02126, 2023.

[18] Yuanhao Gong, Lantao Yu, and Guanghui Yue. Isotropic gaussian splatting for real-time radiance field rendering. *ArXiv*, abs/2403.14244, 2024.

[19] Rui Jin, Yuman Gao, Yingjian Wang, Haojian Lu, and Fei Gao. Gs-planner: A gaussian-splatting-based planning framework for active high-fidelity reconstruction. *ArXiv*, abs/2405.10142, 2024.

[20] Cédric Le Gentil, Othmane-Latif Ouabi, Lan Wu, Cédric Pradalier, and Teresa Vidal-Calleja. Accurate gaussian-process-based distance fields with applications to echolocation and mapping. *IEEE Robotics and Automation Letters*, 9:1365–1372, 2023.

[21] Ho Jin Choi and Nadia Figueroa. Towards feasible dynamic grasping: Leveraging gaussian process distance fields, se(3) equivariance and riemannian mixture models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[22] Kshitij Goel and Wennie Tabib. Distance and collision probability estimation from gaussian surface models. *arXiv preprint arXiv:2402.00186*, 2024.

[23] Silvia Sellan and Alec Jacobson. Stochastic poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 41:1 – 12, 2022.

[24] Seyed Sina Mirrazavi Salehian, Nadia Figueroa, and Aude Billard. A unified framework for coordinated multi-arm motion planning. *The International Journal of Robotics Research*, 37:1205 – 1232, 2018.

[25] Zhiquan Zhang, Tianyu Li, and Nadia Figueroa. Constrained passive interaction control: Leveraging passivity and safety for robot manipulators. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13418–13424, 2024.

[26] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Reed Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Tozeto Ramos, Nathan D. Ratliff, and Dieter Fox. Curobo: Parallelized collision-free robot motion generation. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8112–8119, 2023.

[27] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[28] Adam Fishman, Adithya Murali, Clemens Eppner, Bryan N. Peele, Byron Boots, and Dieter Fox. Motion policy networks. In *Conference on Robot Learning*, 2022.

[29] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

[30] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *European Conference on Computer Vision (ECCV)*, 2018.