

GS-Planner: A Gaussian-Splatting-based Planning Framework for Active High-Fidelity Reconstruction

Rui Jin*, Yuman Gao*, Yingjian Wang, Yuze Wu, Haojian Lu, Chao Xu and Fei Gao

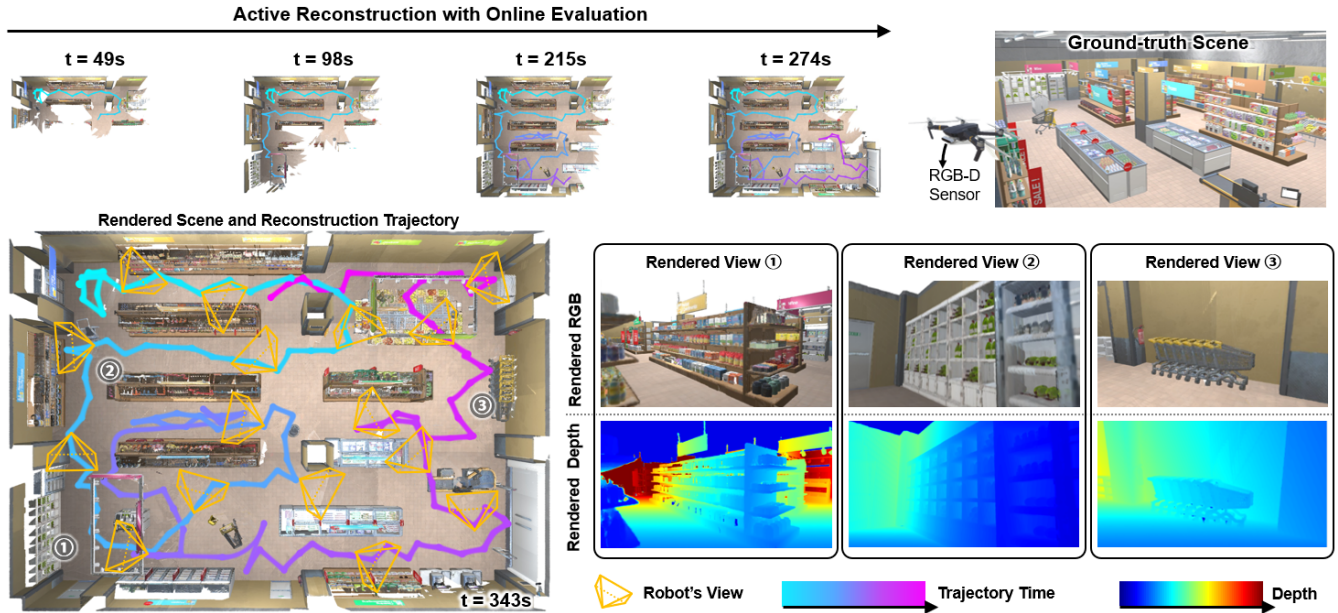


Fig. 1: The whole active reconstruction process in a simulated supermarket scene. We deployed our active high-fidelity reconstruction system on a simulated quadrotor with an RGB-D sensor. The colored curves illustrate the executed trajectories of the drones. We demonstrate the reconstruction result including the whole rendered scene and details rendered at three views.

Abstract—Active reconstruction technique enables robots to autonomously collect scene data for full coverage, relieving users from tedious and time-consuming data capturing process. However, designed based on unsuitable scene representations, existing methods show unrealistic reconstruction results or the inability of online quality evaluation. Due to the recent advancements in explicit radiance field technology, online active high-fidelity reconstruction has become achievable. In this paper, we propose GS-Planner, a planning framework for active high-fidelity reconstruction using 3D Gaussian Splatting. With improvement on 3DGS to recognize unobserved regions, we evaluate the reconstruction quality and completeness of 3DGS map online to guide the robot. Then we design a sampling-based active reconstruction strategy to explore the unobserved areas and improve the reconstruction geometric and textural quality. To establish a complete robot active reconstruction system, we choose quadrotor as the robotic platform for its high agility. Then we devise a safety constraint with 3DGS to generate executable trajectories for quadrotor navigation in the 3DGS map. To validate the effectiveness of our method, we conduct extensive experiments and ablation studies in highly realistic simulation scenes.

This work was supported by the National Key R&D Program of China under grant no. 2023YFB4706600 and the National Natural Science Foundation of China under grant no. 62322314.

***Equal contribution.** All authors are with the Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China, and also with the Huzhou Institute, Zhejiang University, Huzhou 313000, China. {fgaoaa}@zju.edu.cn.

I. INTRODUCTION

Active high-fidelity 3D reconstruction involves robots creating an accurate, detailed, and realistic digital representation of an object or scene completely, efficiently, and safely. Maintaining intricate visual fidelity, this technique demonstrates significant practical value in scene inspection, virtual game development, and cultural heritage preservation.

The choice of an appropriate scene representation is the cornerstone of an active high-fidelity 3D reconstruction robotic system, with the following key requirements:

- **Precision and photorealism:** High-fidelity reconstruction requires the scene representation to accurately represent geometric and textural information, which enables a more realistic portrayal of the scene.
- **Real-time fusion:** New scene information is gathered step by step within the active reconstruction process. The scene representation should fuse the newly collected data in real time to guide the robot's reconstruction and provide information about occupied volumes for safe robot navigation.
- **Online evaluation:** To guide the robot in active reconstruction, the scene representation requires online evaluation of both reconstruction quality and completeness.

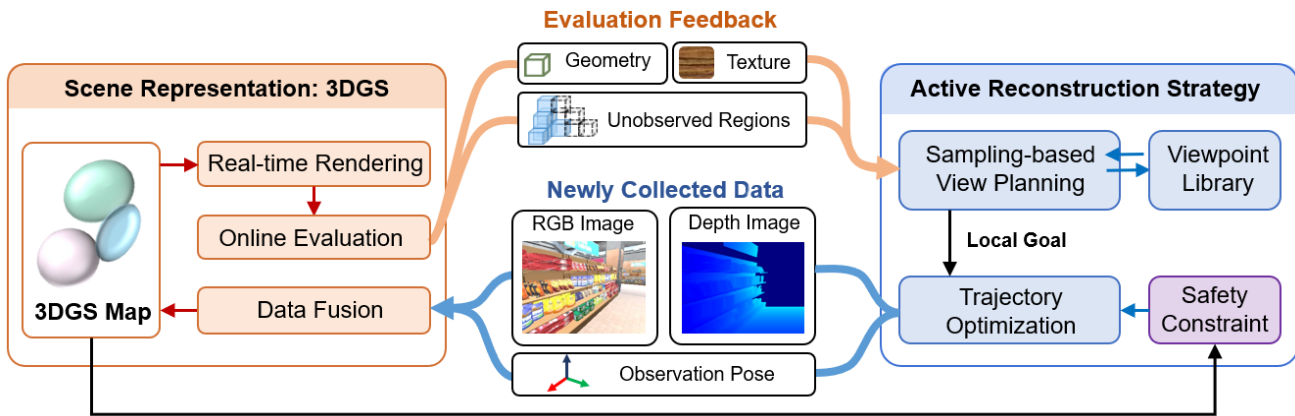


Fig. 2: An overview of our active high-fidelity reconstruction system. With 3DGS as scene representation, the unobserved regions, as well as the geometric and textural information of the built map can be feedback in real time for online reconstruction quality and completeness online evaluation. The proposed active reconstruction strategy guides the robot to collect new scene information to build a complete high-fidelity 3DGS map.

The quality assessment should include both geometric and textural aspects. The completeness evaluation demands the representation to identify observed and unobserved portions of the scene.

However, commonly used in traditional active reconstruction [1]–[4], grid map can only describe coarse structures and lack color and texture information. Mesh and surfel cloud fusion and optimization pose challenges due to their inherent complexities. Neural Radiance Field (NeRF) [5], which recently emerged as a high-fidelity scene representation, often requires extensive training times and substantial resources for rendering, making online evaluation difficult.

3D Gaussian Splatting (3DGS) [6], which recently emerged as a transformative technique in the explicit radiance field, fully meets the above requirements with specific advantages as follows: (a) High visual quality and precise geometry: 3DGS represents a scene with Gaussian blobs storing rich textural and explicit geometric information, ensuring high visual fidelity and precise geometry. More importantly, with learnable 3D Gaussians, 3DGS preserves properties of continuous volumetric radiance fields, which is essential for high-quality image synthesis. (b) Efficient fusion: Benefiting from explicit representation, 3DGS’s frustum culling strategy and adaptive Gaussian densification make it efficient to incrementally fuse the new observed data, showing comparable quality and superior efficiency surpassing neural-based methods. (c) Fast rendering: 3DGS’s highly parallel “splatting” rasterization, along with the avoidance of the computational overhead associated with rendering in empty space, enables high frame rates and high-quality rendering for online evaluation.

Due to 3DGS’s appealing features, we propose a Gaussian-Splatting-based planning framework (GS-planner) to achieve active high-fidelity reconstruction with real-time quality and completeness evaluation to guide the robot’s reconstruction. Firstly, to evaluate the built 3DGS within the reconstruction process, we devise evaluation terms for both reconstruction completeness and quality. Existing 3DGS can only represent

occupied space, making it difficult to evaluate the completeness. To efficiently identify unobserved portions of the scene, we integrate the unknown voxels into the splatting-based rendering process. Secondly, we design a sampling-based active exploration strategy to guide the robot to explore the unobserved areas and improve the geometric and textural quality of the 3DGS map. Thirdly, to form a complete robotic active reconstruction system, we select quadrotor as the robotic platform for its high agility. Leveraging the differentiable nature and explicit representation properties of 3DGS, we devise a differentiable obstacle-avoidance cost with the 3DGS map. Furthermore, we form an autonomous navigation framework capable of generating collision-free and dynamic-feasible trajectories for quadrotors. Overall, based on the state-of-the-art dense 3DGS SLAM system SplaTam [7], we propose GS-Planner, a planning framework for active high-fidelity reconstruction with 3DGS as scene representation. In summary, the following are the contributions:

- 1) We propose the first active 3D reconstruction system using 3DGS with online evaluation.
- 2) We design evaluation metrics for reconstruction completeness and quality, applying them in a sampling-based autonomous reconstruction framework.
- 3) We devise a safety constraint with 3DGS and form a trajectory optimization framework in the 3DGS map.
- 4) We conduct extensive simulation experiments to validate the effectiveness of the proposed system.

II. RELATED WORKS

A. High-fidelity Reconstruction

To achieve high-fidelity reconstruction, several different scene representations have been employed, such as planes, meshes, and surfel clouds. Recently, Neural Radiance Field (NeRF) [5] has gained prominence in this field due to its exceptional capability of photorealistic rendering, which can be divided into three main types: MLP-based methods,

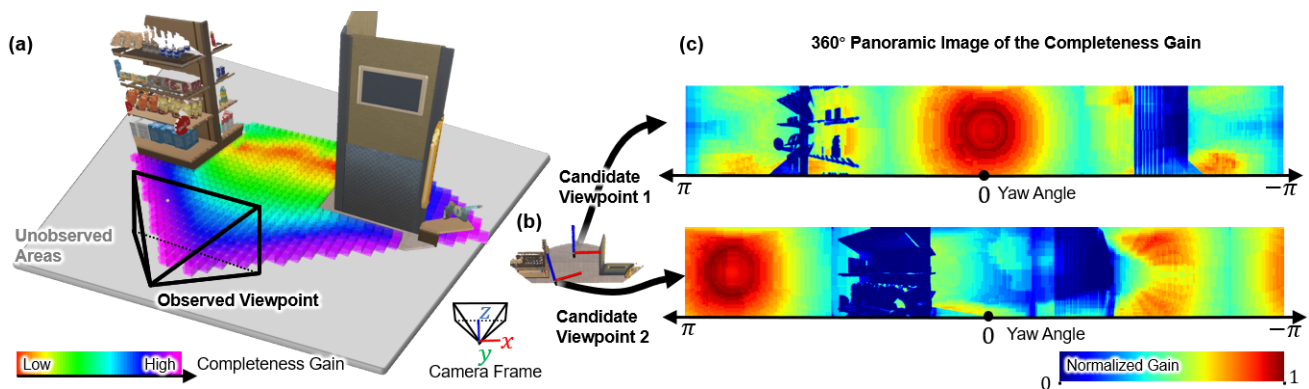


Fig. 3: An illustration of the completeness evaluation. (a). A partially reconstructed scene. Scene information has been collected only at the observed viewpoint. The colored grid illustrates the completeness gain from 360-degree summation at different positions at a height of $z = 1m$. (b). The location of two candidate viewpoints. The z-axis direction is aligned with the camera’s optical axis. (c). 360-degree panoramic image of the completeness gain of the candidate viewpoint 1 and 2. The generation of 360-degree gain facilitates the subsequent determination of the optimal viewpoint yaw direction.

hybrid representation methods, and explicit methods. MLP-based method [8] offers scalable and memory-efficient map representations but faces challenges with catastrophic forgetting in larger scenes. Hybrid representation [9, 10] methods combine the advantages of implicit MLPs and structure features, significantly enhancing the scene scalability and precision. As for the explicit method proposed in [11], it stores map features in voxel directly, without any MLPs, enabling faster optimization.

While NeRF excelled in photorealistic reconstruction [12], NeRF methods are computationally intensive [13]–[15]. NeRF often requires extensive training times and substantial resources for rendering, which contradicts the need to feed the model back into the active reconstruction decision loop in real time. Instead of representing maps with implicit features, 3DGS [6] enables real-time rendering of novel views by its pure explicit representation and the novel differential point-based splatting method. This technology has been applied in online dense SLAM with 3DGS as the scene representation and reconstructs the scene from RGB-D images [7, 16].

B. Active Reconstruction System

Active reconstruction systems put data acquisition in the decision loop, using the results for evaluation, and then guiding the robot for further data acquisition. Based on the representations of 3D models, these approaches can be categorized into voxel-based methods [1]–[4], surface-based approaches [17]–[19], and neural-based approaches [12].

Voxel-based methods [1, 3, 4] aim to reconstruct the commonly used grid map, which is an axis-aligned and compact spatial representation. Surface-based approaches [17]–[19] model the environment with a set of surfaces. However, these methods only evaluate the reconstruction completeness but ignore color and texture information. There are also active reconstruction methods based on implicit neural representations. NeurAR [12] learns the neural uncertainty for view planning. However, limited by the high computation consumption of the implicit neural representation,

NeurAR takes about 50-120 seconds for model optimization and uncertainty evaluation between view steps, leading to frequent and prolonged halts in robot operation. 3DGS, as a newly emerged method, is well-suited for serving as a scene representation for active high-fidelity reconstruction. However, there is currently no active reconstruction robot system designed based on its excellent characteristics.

III. SYSTEM OVERVIEW

Active high-fidelity reconstruction requires a robot to visit a series of viewpoints to collect scene information and build a realistic digital representation. As shown in Fig. 2, the proposed active reconstruction system uses 3DGS as scene representation, and the robot can collect RGB-D images with the corresponding observation poses. Leveraging the efficient fusion and real-time rendering advantages of 3DGS, we conduct an online evaluation for possible future viewpoints. Such online evaluation feedback guides the active view planning module (Sec. IV) to generate a series of safe and high-information-gain viewpoints. To navigate the robot to the selected viewpoints, we further propose an autonomous navigation framework (Sec. V) with a safety constraint formulated with the 3DGS map.

IV. ACTIVE VIEW PLANNING WITH 3DGS MAP

In this section, we first introduce the 3DGS representation (Sec. IV-A). Then, a completeness evaluation method (Sec. IV-B) and a quality evaluation method (Sec. IV-C) are proposed to capture regions with poor coverage and quality respectively. In the following, we design a sampling-based active view planning algorithm to guide the robot to reconstruct unobserved regions and improve the quality of the built map (Sec. IV-D).

A. 3DGS Map Representation

We use the existing method SplaTam SLAM [7] for 3DGS real-time reconstruction. The scene is represented as a set of isotropic 3D GS. Each 3D GS is defined by center position

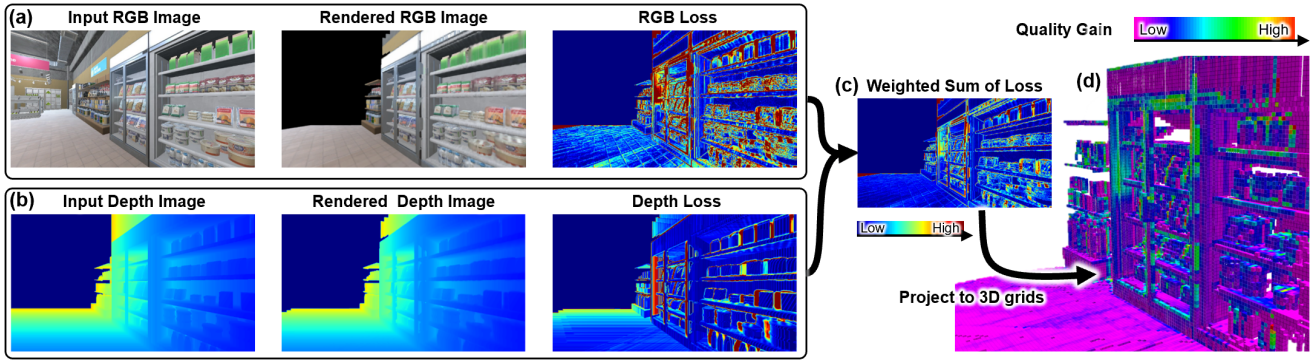


Fig. 4: An instance of the quality evaluation. (a). The generation of the RGB textural loss between the input RGB image and rendered RGB image. (b). The generation of the depth loss between the input depth image and rendered depth image. (c). The weighted sum of the RGB loss and depth loss. (d). We project the quality gain to the 3D grid in the world frame to store.

$\mu \in \mathbb{R}^3$, radius $r \in \mathbb{R}$, RGB color $c \in \mathbb{R}^3$, and opacity $o \in \mathbb{R}$. The opacity function α of a point $x \in \mathbb{R}^3$ computed from each 3DGS is described as:

$$\alpha(x, o) = o \exp\left(-\frac{|x - \mu|^2}{2r^2}\right). \quad (1)$$

In order to optimize the parameters of 3D Gaussians to represent the scene, we need to render them into images in a differentiable manner. The final rendered color can be formulated as the alpha-blending of N ordered points that overlap the pixel,

$$C_{pix} = \sum_{i=1}^N c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (2)$$

We render the depth in the same way

$$D_{pix} = \sum_{i=1}^N d_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (3)$$

where d_i represents the depth of the i -th 3D Gaussian's center, which is equal to the z -coordinate of its center position in camera coordinate system.

B. Completeness Evaluation

To support full coverage of the scene, we introduce the completeness evaluation for candidate viewpoints. This evaluation requires to recognize unobserved space. However, the existing 3DGS only preserves information regarding the occupied volume. To address this limitation, we maintain a voxel map to represent unobserved volume, and integrate it into the splatting-based rendering. Then, we can efficiently calculate model-consistent pixel-level completeness gain within the 3DGS rendering process.

To be specific, given a collection of 3D Gaussians and a candidate viewpoint, first all Gaussians will be sorted by their depth. With the sorted Gaussians, depth image can be efficiently rendered by alpha-compositing the splatted 2D projection of each Gaussian in order in pixel space. In this rendering process, we can determine whether there exists an unobserved region between adjacent Gaussians

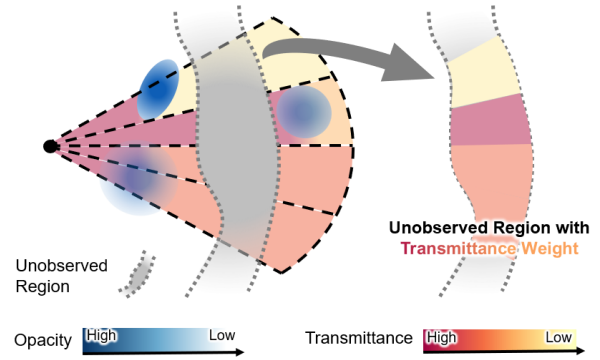


Fig. 5: A 2D illustration of the 3D completeness evaluation. Given a collection of 3D Gaussians and a candidate viewpoint, we can get unobserved regions within the splatting-based rendering. The unobserved regions are weighted by transmittance, which is equal to the accumulated Gaussians' opacity along the ray.

utilizing the maintained unobserved voxel. And the volume of the unobserved region corresponding to each pixel can be approximately calculated by the basic frustum volume formula. Furthermore, considering that Gaussians have different opacities, we evaluate the visibility of the unobserved volume by applying a transmittance weight, as shown in Fig. 5. Finally, we get the completeness information gain of each pixel as

$$V_{pix} = \sum_{i=1}^n V_i \prod_{j=1}^{m_i} (1 - \alpha_j), \quad (4)$$

where n is the number of unobserved volumes along the ray, m_i is the number of the related 3D Gaussians before the i -th unobserved volume V_i , $\prod_{j=1}^{m_i} (1 - \alpha_j)$ is the transmittance weight. For a certain unobserved volume V_i , as shown in Fig. 6, we approximate its volume as a frustum:

$$V_i = \frac{1}{3} (S_{in,i} + \sqrt{S_{in,i} S_{out,i}} + S_{out,i}) (d_{out,i} - d_{in,i}), \quad (5)$$

where $d_{in,i}$ and $d_{out,i}$ respectively represent the depths of the entry and exit of the i -th unobserved volume. $S_{in,i}$ and $S_{out,i}$ represent the base areas of V_i , which are equal to the projected areas of the pixel at the entry plane of depth $d_{in,i}$

and the exit plane of depth $d_{out,i}$. $S = d^2/f^2$, where f is the camera focal length.

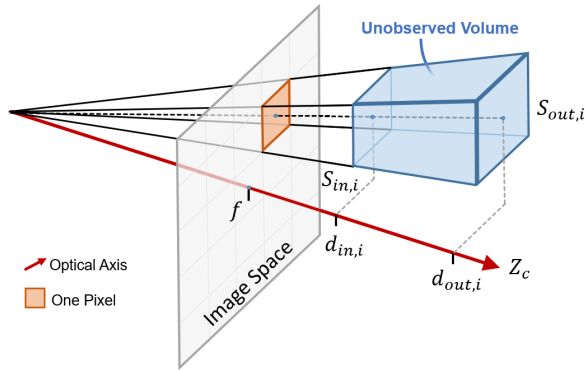


Fig. 6: A 3D illustration of the unobserved volume calculation.

Because we integrate the evaluation into the splatting-based rendering, this calculation process is parallelized and efficient. To illustrate the completeness evaluation intuitively, we give an instance shown in Fig. 3, demonstrating the guidance of completeness evaluation in viewpoint selection.

C. Quality Evaluation

Quality evaluation aims to identify reconstructed regions with poor texture and geometry accuracy. This evaluation includes two steps: loss caching and loss reprojection.

Loss caching: Leveraging the real-time rendering of 3DGS, it is straightforward to compute the disparity between the reconstructed model and the actual scene. As shown in Fig. 4, we project the loss L from the image space to the world space, and cache the loss into occupied voxels. Specifically, L is a weighted sum of L_1 loss both on the depth and the color renders:

$$L = L_1(D) + \lambda_C L_1(C), \quad (6)$$

where λ_C is the weight coefficient.

Loss reprojection: Given a candidate viewpoint, we reproject the loss cached in the occupied voxels to the image space by conducting 360-degree ray-tracing. The loss indicates the quality information gain of both texture and geometry.

According to Sec. IV-B and Sec. IV-C, we finally obtain an overall 360-degree information gain of a given viewpoint by calculating the weighted sum of completeness and quality. Then we use the sliding-window summation to find the optimal yaw angle of each viewpoint.

D. View Planning with a View Library

To enable a robot's full reconstruction of a scene in high quality, a series of reasonable viewpoints with position and yaw angle need to be generated for sequential navigation. We design a sampling-based view planning method with a view library [4] to generate and cache viewpoints for evaluation. The whole view planning algorithm is listed as Alg. 1.

To be specific, we first acquire nearby cached viewpoints \mathbf{V}_{near} in the view library \mathbf{VL} , which stores unvisited viewpoints and their information gain (Line 1). Their information

Algorithm 1 Active View Planning with a View Library

Require: current pose \mathbf{p} , view library \mathbf{VL} ;

- 1: $\mathbf{V}_{near}, \mathbf{G}_{near} \leftarrow$ subset of \mathbf{VL} nearby current pose \mathbf{p} ;
- 2: **for** $v_i \in \mathbf{V}_{near}, g_i \in \mathbf{G}_{near}$ **do**
- 3: $g_i = \text{UpdateGain}(v_i)$;
- 4: **end for**
- 5: $\mathbf{V}_c \leftarrow \text{RRTSample}(\mathbf{p}, \mathbf{V}_{near})$;
- 6: $\mathbf{G}_c \leftarrow \text{Evaluation}(\mathbf{V}_c)$;
- 7: **for** $v_i \in \mathbf{V}_c, g_i \in \mathbf{G}_c$ **do**
- 8: **if** $g_i < g_{lb}$ **then**
- 9: $\mathbf{V}_c \leftarrow \mathbf{V}_c \setminus v_i$;
- 10: $\mathbf{VL} \leftarrow \mathbf{VL} \setminus \{v_i, g_i\}$;
- 11: **continue**;
- 12: **end if**
- 13: **if** $\text{Overlap}(v_i, \mathbf{V}_{near}) < \epsilon_{ol}$ **and** $g_i > g_{thres}$ **then**
- 14: $\mathbf{VL} = \mathbf{VL} \cup \{v_i, g_i\}$;
- 15: **end if**
- 16: **end for**
- 17: Result local goal: \mathbf{p}_{goal} ;
- 18: **if** $\mathbf{V}_c \neq \emptyset$ **then**
- 19: $\mathbf{p}_{goal} = \text{BestBranchNode}(\mathbf{V}_c)$;
- 20: **else**
- 21: $\mathbf{p}_{goal} = \text{Select from } \mathbf{VL}$;
- 22: **end if**
- 23: **Return** \mathbf{p}_{goal} ;

gains are updated with new sensor data (Line 2-4). We use the expansion part of RRT* to sample potential future viewpoints \mathbf{V}_c (Line 5). The sampled viewpoints that are too close to obstacles will be deleted. And the optimal yaw angle of each viewpoint is determined by the above introduced sliding window method. \mathbf{V}_{near} are added and connected to the expanded trees in the sampling process. By real-time rendering at each viewpoint in \mathbf{V}_c via 3DGS, we calculate its information gain efficiently (Line 6). The viewpoints whose gain below threshold g_{lb} will be removed (Line 8-12). And high-information-gain viewpoints that are novel enough from others in \mathbf{VL} will be cached (Line 13-15). The node on the best branch will be selected as the next local goal (Line 19). Moreover, if there are no valid nearby candidates, the local goal will be selected from \mathbf{VL} (Line 21). When the \mathbf{VL} becomes empty, the reconstruction process is accomplished.

V. TRAJECTORY OPTIMIZATION IN 3DGS MAP

3DGS's explicit representation and precise geometry make safe robot navigation with the 3DGS map possible. Leveraging the differentiable nature of 3D Gaussian, we devise a safety constraint with the 3DGS map, and integrate it into a quadrotor trajectory optimization framework.

A. Safety Constraint with 3DGS

In 3DGS, Gaussians are defined with opacity as presented in Sec. IV-A. The opacity measures the probability of light being obstructed while passing through an object. We assume

the probability of terminating a light ray provides a strong indication of the probability of terminating a mass particle. Thus, for a robot pose p and a certain Gaussian with opacity o , we formulate a chance constraint to ensure safety:

$$\alpha(p, o) < c_{thr}, \quad (7)$$

where $\alpha(\cdot)$ presents the opacity function defined in Eq. 1, and c_{thr} presents the threshold of collision probability. c_{thr} is equal to the value of $\alpha(\cdot)$ at a distance of $(3r + R_{robot})$ (3σ rule) from its mean μ when the opacity $o = 1$:

$$c_{thr} = \exp\left(-\frac{3r + R_{robot}}{2r^2}\right), \quad (8)$$

where R_{robot} is the geometric bounding sphere radius. Intuitively, it means that we hope every point on the trajectory is at a distance greater than a safety radius R_s from the Gaussian mean point. R_s is weighted by o of the Gaussian, and equals to $(3r + R_{robot})$ when $o = 1$.

For the follow-up trajectory optimization, we provide the corresponding collision-avoidance cost for each point p on the trajectory as

$$\mathcal{J}_c(p) = \sum_{i=0}^k f(\alpha_i(p, o_i) - c_{thr}), \quad (9)$$

where $f(x) = \max(x, 0)^3$, and k is the number of nearby Gaussian elements in the 3DGS map. The collision-avoidance cost applied on the points on the trajectory during optimization with different opacity Gaussian is shown in Fig. 7. This differentiable cost is friendly for the follow-up trajectory optimization with analytical gradient written as

$$\frac{\partial \mathcal{J}_c(p)}{\partial p} = \sum_{i=0}^k 3(\alpha_i(p) - c_{thr})^2 o_i \exp\left(-\frac{|p - \mu_i|^2}{2r_i^2}\right) \left(\frac{\mu_i - p}{2r_i^2}\right). \quad (10)$$

B. Trajectory Optimization Formulation

Aimed to generate full-state collision-free and dynamic-feasible trajectories for quadrotors, we use MINCO [20]

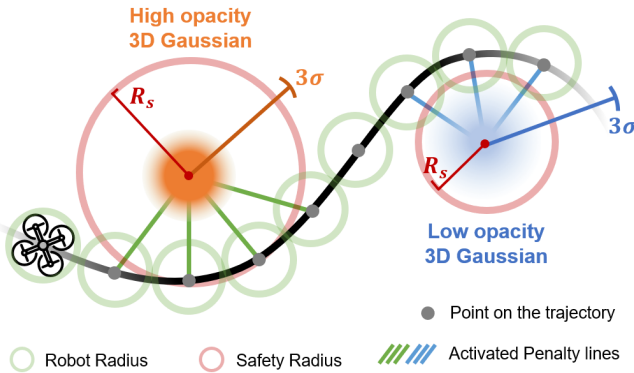


Fig. 7: The collision-avoidance cost applied on the trajectory with different opacity Gaussian. Each point on the trajectory is hoped to be at a distance greater than a safety radius R_s from the mean point of the Gaussian. R_s is weighted by the opacity o of different Gaussians.

as trajectory representation and optimize spatial-temporal trajectories in a reduced space with differential-flat outputs $\mathbf{z} = [\mathbf{p}^T, \phi]^T \in \mathbb{R}^3 \times \text{SO}(2)$, where ϕ is the Euler-yaw angle and position $\mathbf{p} = [p_x, p_y, p_z]^T$. And we further define the flat outputs and their derivatives $\mathbf{z}^{[s-1]} \in \mathbb{R}^{ms}$ as $\mathbf{z}^{[s-1]} := (\mathbf{z}^T, \dot{\mathbf{z}}^T, \dots, \mathbf{z}^{(s-1)T})^T$. To generate a trajectory $\mathbf{z}(t) : [0, T] \mapsto \mathbb{R}^m$, we formulate the trajectory optimization problem as

$$\min_{\mathbf{z}, \mathbf{T}} \mathcal{J}_E = \int_0^T \|\mathbf{z}^{(s)}(t)\|^2 dt + \rho T, \quad (11a)$$

$$s.t. \quad \mathbf{z}^{[s-1]}(0) = \bar{\mathbf{z}}_s, \quad \mathbf{z}^{[s-1]}(T) = \bar{\mathbf{z}}_e, \quad (11b)$$

$$\|\mathbf{p}^{(1)}(t)\| \leq v_{max}, \quad \forall t \in [0, T], \quad (11c)$$

$$\|\mathbf{p}^{(2)}(t)\| \leq a_{max}, \quad \forall t \in [0, T], \quad (11d)$$

$$\|\phi^{(1)}(t)\| \leq \phi_{max}, \quad \forall t \in [0, T], \quad (11e)$$

$$\alpha_i(\mathbf{p}, o_i) < c_{thr}, \quad \forall i \in \{1, \dots, k\}, \quad \forall t \in [0, T], \quad (11f)$$

where Eq. 11a trade off the smoothness and aggressiveness, and ρ is the time regularization parameter. Here we adopt $s = 3$ for jerk integral minimization. Eq. 11b is the boundary conditions at start and end time. $\bar{\mathbf{z}}_s$ and $\bar{\mathbf{z}}_e$ are the initial and end state, respectively. Eq. 11c, Eq. 11d and Eq. 11e are the dynamic feasibility constraints, where v_{max} , a_{max} and ϕ_{max} are the velocity, acceleration and yaw rate limits. Eq. 11f is the safe constraint defined in Eq. 7. $\alpha_i(\cdot)$ is the opacity function of the i -th Gaussian element with opacity o_i .

This problem can be transformed into an unconstrained optimization problem [20] written as

$$\min_{\mathbf{z}, \mathbf{T}} \mathcal{J}_E + \int_0^T \mathcal{J}_G dt, \quad (12)$$

where \mathcal{J}_G is the penalty function corresponding to the inequality constraints Eq. 11c, Eq. 11d, and Eq. 11f. And \mathcal{J}_G includes \mathcal{J}_c defined in Eq. 10. With analytical gradients, the problem is then efficiently solved by the L-BFGS [21].

VI. EXPERIMENTS

A. Implementation Details

We run our active reconstruction system on a desktop PC with a 2.90GHz Intel i7-10700 CPU and an NVIDIA RTX 3090 GPU. And an additional laptop PC with a 2.50 GHz AMD Ryzen 9 7945HX and an NVIDIA GeForce RTX 4080 Laptop GPU is utilized to execute the high-fidelity simulation developed with Unity. The two devices are connected via a wired network connection. In Unity, the quadrotor equipped with an RGB-D sensor will provide real-time RGB-D images with a resolution of 640×480 and a perceptual range from $0.5m$ to $3m$. We add a uniform distribution noise of $2cm$ to the depth and assume the corresponding camera poses of the images are known.

The 3DGS mapping module builds upon SplaTam [7] by incorporating a real-time data streaming format. For view planning, we evaluate 10 viewpoints at each iteration and select the branch with the optimal viewpoint as the seed for the next iteration. For trajectory optimization, the robot

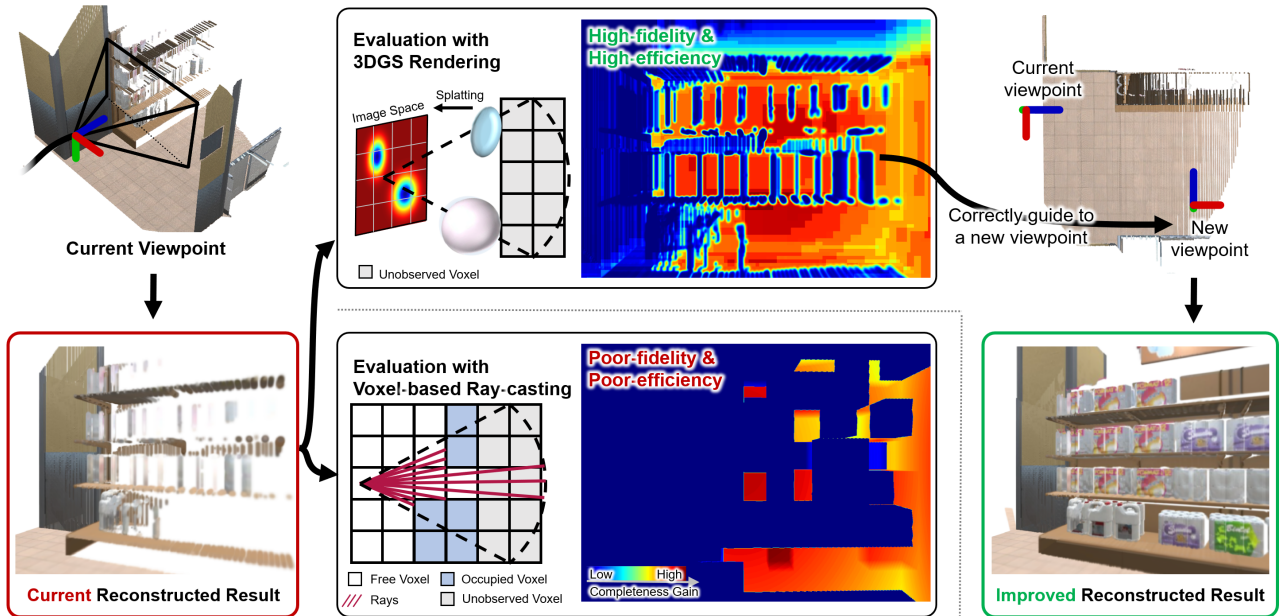


Fig. 8: Qualitative comparison of the completeness evaluation between using 3DGS rendering and using voxel-based ray-casting. When the robot arrives at the current viewpoint, due to its oblique view angle, the observation of items on the left front shelf is incomplete. The evaluation with 3DGS rendering is high-fidelity and high-efficiency, while the evaluation with voxel-based ray-casting is coarse and time-consuming. The fine completeness evaluation can correctly guide the robot to collect new information for improvement.

radius is fixed at 0.5m. And the safety constraint is computed by considering the 3DGS near the initial trajectory within the duration of $[0s, 1s]$, selected using the Axis-aligned Bounding Box (AABB) method. The maximum velocity limit is $1.0m/s$, the maximum acceleration limit is $2.0m/s^2$, and the maximum yaw rate limit is $\pi rad/s$.

B. Simulation Result and Analysis

To validate our proposed method, we build a high-fidelity simulation environment via Unity engine. As shown in Fig. 1, this $22.0m \times 14.0m \times 3.2m$ supermarket scene contains a variety of items with rich texture information. We present the whole reconstruction process and the trajectory of the quadrotor. The quadrotor takes 343 seconds to complete the whole reconstruction. The reconstructed details are also demonstrated through rendered RGB and depth images. We can see from the reconstruction results that the reconstruction of the entire scene is complete and high-fidelity, retaining rich texture and structural information, and exhibiting a strong sense of realism.

C. Comparison and Ablation Study

To validate the effectiveness of the proposed reconstruction evaluations, we compare our method with traditional ones and conduct an ablation study.

1) *Completeness Evaluation*: Given a viewpoint, traditional methods for computing information gain typically rely on voxel-based raycasting [1]–[3]. This involves maintaining a grid map that represents observed and unobserved areas, and performing raycasting at candidate viewpoints to measure the volume of unobserved areas. However, this method is limited by the voxel resolution for occupied

TABLE I: Completeness Evaluation Methods Comparison

Voxel Resolution (m)	Scenario	Time (ms)	
		Voxel-based Raycast	Ours
0.1	Sparse	347.32	1.86
	Dense	342.10	2.11
0.15	Sparse	226.29	1.83
	Dense	230.21	2.33
0.2	Sparse	183.36	1.71
	Dense	176.01	2.31

and unobserved region representation, and its computational complexity is affected by discrete sampling steps. In contrast, we integrate the completeness evaluation calculation into the splatting process. Leveraging efficient Gaussian sorting and precise description of occupied geometry, we achieve high-fidelity high-efficiency completeness gain calculation. Fig. 8 shows an instance of the completeness gain calculation by different methods. Tab. I compares the computation speeds under various voxel resolutions, highlighting the notably higher efficiency of our 3DGS-based method. In the experiments, the raycasting step for the voxel-based method is half of the voxel resolution.

2) *Quality Evaluation*: To validate the impact of quality gain, we designed ablation experiments to calculate the information gain of candidate viewpoints with and without quality gain. And we further compute their corresponding optimal yaw angles. As the result shown in Fig. 9, the quality gain correctly guides the generation of the information gain and the optimal yaw angle. With quality consideration, our active reconstruction system can improve the regions of the built scene with poor geometry and texture.

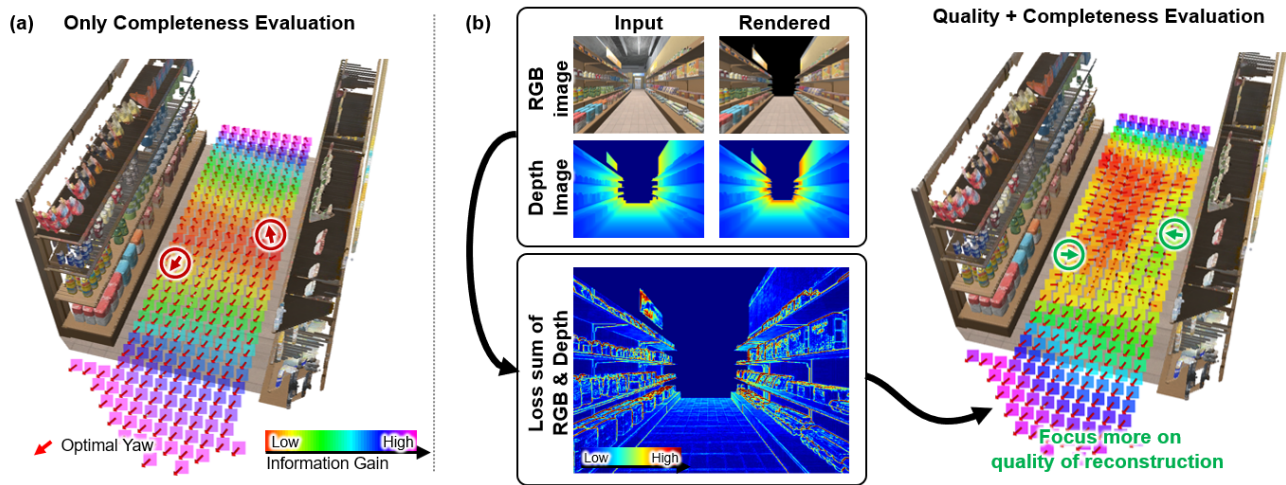


Fig. 9: Ablation of the quality gain. (a). The information gain regarding only completeness at the height of $z = 1$ m. Optimal yaw angles corresponding to the candidate viewpoints point towards unobserved areas. (b). Considering both quality and completeness in the information gain. It can be observed that, for viewpoints around two shelves, the quality gain tends to encourage further observation of shelves that can still improve the reconstruction quality.

VII. CONCLUSION AND FUTURE WORK

In this paper, we adopt the recently emerged 3DGS technique to achieve an active high-fidelity reconstruction system. To online evaluate the reconstruction result as reconstruction strategy feedback, we respectively design completeness and quality evaluation methods with 3DGS. Then we propose a sampling-based active view planning method to generate a series of optimal viewpoints. For robot navigation in 3DGS map, we design a differentiable chance constraint to ensure safety, and form a quadrotor trajectory optimization framework. For future work, we are going to deploy our system on real robotic platforms and try to reduce the GPU memory consumption of 3DGS and improve its efficiency.

REFERENCES

- [1] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3d reconstruction," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3477–3484.
- [2] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3105–3112.
- [3] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.
- [4] M. Corah, C. O’Meadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1715–1721, 2019.
- [5] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [6] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, 2023.
- [7] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat, track & map 3d gaussians for dense rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [8] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.
- [9] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.
- [10] C. Jiang, H. Zhang, P. Liu, Z. Yu, H. Cheng, B. Zhou, and S. Shen, "H2-mapping: Real-time dense mapping using hierarchical hybrid representation," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6787–6794, 2023.
- [11] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.
- [12] Y. Ran, J. Zeng, S. He, J. Chen, L. Li, Y. Chen, G. Lee, and Q. Ye, "Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1125–1132, 2023.
- [13] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler, "Neural geometric level of detail: Real-time rendering with implicit 3d shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 358–11 367.
- [14] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 335–14 345.
- [15] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–15, 2022.
- [16] C. Yan, D. Qu, D. Wang, D. Xu, Z. Wang, B. Zhao, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," *arXiv preprint arXiv:2311.11700*, 2023.
- [17] R. Huang, D. Zou, R. Vaughan, and P. Tan, "Active image-based modeling with a toy drone," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6124–6131.
- [18] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.
- [19] Y. Gao, Y. Wang, X. Zhong, T. Yang, M. Wang, Z. Xu, Y. Wang, Y. Lin, C. Xu, and F. Gao, "Meeting-merging-mission: A multi-robot coordinate framework for large-scale communication-limited exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 700–13 707.
- [20] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, 2022.
- [21] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.