2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Efficient View Path Planning for Autonomous Implicit Reconstruction

Jing Zeng[1] Yanxu Li[1] Yunlong Ran[1] Shuo Li[1] Fei Gao[1] Lincheng Li[2] Shibo He[1] Jiming Chen[1] Qi Ye[1]

***Abstract*— Implicit neural representations have shown promising potential for 3D scene reconstruction. Recent work applies it to autonomous 3D reconstruction by learning information gain for view path planning. Effective as it is, the computation of the information gain is expensive, and compared with that using volumetric representations, collision checking using the implicit representation for a 3D point is much slower. In the paper, we propose to 1) leverage a neural network as an implicit function approximator for the information gain field and 2) combine the implicit fine-grained representation with coarse volumetric representations to improve efficiency. Further with the improved efficiency, we propose a novel informative path planning based on a graph-based planner. Our method demonstrates significant improvements in the reconstruction quality and planning efficiency compared with autonomous reconstructions with implicit and explicit representations. We deploy the method on a real UAV and the results show that our method can plan informative views and reconstruct a scene with high quality.**

I. Introduction

In this paper, we study the problem of view path planning for a mobile robot to reconstruct high quality 3D models of unknown scenes. The robot is required to autonomously plan and execute a path that maximizes the quality of the reconstructed 3D scenes. Autonomous reconstruction has wide applications [4], [26], [27], such as virtual reality, digital twin, autonomous driving, and smart city etc..

Recently, implicit neural representations have shown compelling results in 3D scene reconstruction [1], [13], [24] and promising potentials in SLAM [23], [28]. Ran et al. [15] apply an implicit representation, represented by a multilayer perceptron (MLP) $\mathbf{F}_\theta$, to autonomous 3D reconstruction by learning information gain for the view path planning. Effective as it is in reconstructing fine-grained 3D scenes with high fidelity, calculating the information gain field for different viewpoints from the representation is inefficient: multiple rays (R rays) are required to cast through the scene and multiple points (N points) are sampled on each ray to integrate the reconstruction uncertainty for the frustum covered by each viewpoint; all these points are passed to the MLP $\mathbf{F}_\theta$ to estimate the uncertainties. R scales with the image size and N scales with the scene size and the level of details required for the reconstruction quality. The other limitation is the inefficiency of collision checking. To check whether 3D point is free or not, the point is fed into $\mathbf{F}_\theta$ to get the density value while in explicit volumetric representations, only the memory for the point is queried.

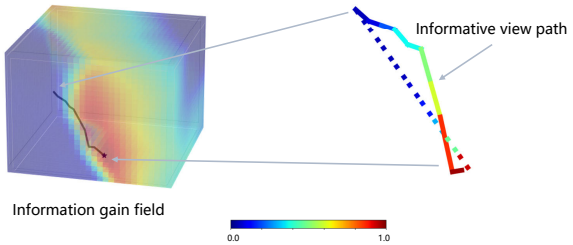

Fig. 1: Left: An information gain approximator $\mathbf{g}_\phi$ fitted for a local gain field at the fifth step of the autonomous implicit reconstruction for the *cabin* scene. Right: an informative view path planned by our method (the solid line) and the shortest path planned by A* (the dashed line). The path is colored with the information gain.

On the other hand, as the computation of the information gain for viewpoints is expensive, sampling-based methods e.g. RRT or RRT* are preferred in previous view path planning work [8], [16], [18], [20] as they typically require fewer number of sampled viewpoints, though many of them are not guaranteed to be optimal.

To address the above limitations, we propose an efficient view path planning for the autonomous reconstruction based on the novel implicit representations. Firstly, we assume the information gain field to be a smooth continuous function of the viewpoints and in the same spirit of the implicit scene representation, we propose to leverage a MLP $\mathbf{g}_\phi$ as an approximator for the field. With the approximation, getting the information gain of a viewpoint requires only one query of $\mathbf{g}_\phi$, instead of querying $\mathbf{F}_\theta$ for the scene radiance field for about a million times. Secondly, volumetric representations are introduced to complement to the implicit representations: the viewpoints to be sampled for the fitting of $\mathbf{g}_\phi$ are filtered based on a coarse TSDF to further reduce the computation of the information gain; an occupancy map from the TSDF is built for the fast collision checking. Thirdly, with the improved efficiency, the query of the information gain for a viewpoint is reduced to less than a millisecond, which opens the possibility of using planners providing almost optimal paths but requiring dense queries of the space for view path planning. Therefore, we demonstrate the possibility with a A* planner and propose a novel view path cost based on it.

To summarize, our contributions are:

- We propose an implicit function approximator for the information gain field, which reduces the time complex-

[1]Zhejiang University, Hangzhou, 310027, China.
[2] Fuxi AI Lab, NetEase, Hangzhou, 310052, China.
Author Jing Zeng (zengjing@zju.edu.cn) and Corresponding Author Qi Ye (qi.ye@zju.edu.cn). Qi Ye is with the College of Control Science and Engineering, the State Key Laboratory of Industrial Control Technology, Zhejiang University, and also the Key Laboratory of Collaborative Sensing and Autonomous Unmanned Systems of Zhejiang Province.

This work was supported in part by NSFC under Grants 62233013, 62088101, and the Fundamental Research Funds for the Central Universities.

ity of querying the information gain for a viewpoint by at least $R \times N$ times.

- We propose a combination of implicit representations for fine-grained 3D scene reconstruction and volumetric representations for fast collision checking and viewpoints filtering.
- We propose a novel view path cost based on a graph based planner, which plans the shorter view paths and provides the better reconstruction quality than existing sampling-based planners.

II. Related work

The problem of determining the optimal viewpoints and paths for efficiently reconstructing a scene is known as the active vision or view-path-planning problem. This problem has been extensively studied for two decades [2], [3], [5], [8], [17], [22]. The most popular methods to solve the problem are frontier-based [9], [18] and sampling-based methods [16], [20]. Frontier-based methods focus on the boundaries between known and unknown space to complete fast and global exploration tasks but are difficult to adapt to other tasks. Sampling-based methods focus on a Next-Best-View (NBV) strategy which selects NBV using feedback from the current partial reconstruction. NBV is determined using information gains of viewpoints which is defined over volumetric representations [6], [7], [11] or surface-based representations [16], [19], [25].

With the information gains of viewpoints defined, sampling-based planners are most commonly used to plan an informative view path. Rapidly Exploring Random Tree (RRT) is exploited to randomly sample viewpoints as nodes in RRT and use edges to connect viewpoints [8], [11], [16]. Mendez et al. [11] use Sequential Monte Carlo to assign sampling weights in different locations for the information gain. A node with the maximum information gain is selected as the target node and the robot is guided along the tree to it. The method requires many sampling viewpoints for the convergence of Sequential Monte Carlo (SMC). Schmid et al. [16] instead keep the entire tree alive and rewire the RRT after every planning iteration to reduce unnecessary re-computation of information gain. However, the rewiring process is time-consuming. Kompis et al. [8] use an artificial potential field to predict the value of proposed viewpoints to save the computation time of calculating their actual information gain. However, finding the frontiers of the surface is not efficient and predicting the value of viewpoints by manually defined parameters is not robust.

Due to the computational inefficiency of the information gain, planning methods sample a small subset of the viewpoints and focus on the design of reducing the number of sampling points for the information gain query. Our method, instead, focuses on improving the computation efficiency of the information gain. The super efficient querying of the information gain field then opens the possibility of investigating graph-based planning methods or other methods, which gives better optimality but requires the dense queries of the space for the view path planning problem.

III. Method

The problem considered is to generate a trajectory for a robot that yields high-quality 3D models of a bounded target scene and fulfills robot constraints like time and path length. The trajectory is composed of a sequence of paths and viewpoints $\Omega = (\omega_1 ..., \omega_n)$ where $\omega_i \in \mathcal{R}^3 \times \mathcal{SO}(2)$. Finding the best sequence of viewpoints V is time-consuming and prohibitively expensive. Similar to [15], [16], we adopt the greedy strategy and treat it as a Next-Best-View (NBV) problem. At each step of the reconstruction, the information gain of viewpoints within a certain range of the current position of a robot are evaluated based on the partial reconstruction scene. An informative path considering both the information gain and the path length is then planned and executed. Images captured along the view path are selected and are fed into the 3D reconstruction of next step. The process is repeated until some critera are met.

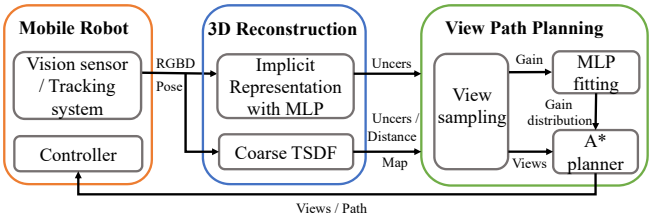

Fig. 2: The pipeline of our proposed method.

A. System Overview

Under the greedy strategy, our pipeline consists of three components as can be seen in Fig. 2. The Mobile Robot module takes the images at given viewpoints and the robot locates itself by a motion capture system. During the simulation, Unity Engine renders images at given viewpoints. The 3D Reconstruction module reconstructs a scene by combining an implicit neural representation (e.g. NeRF [12]) and a volumetric representation (A coarse TSDF). The implicit neural representation provides high quality 3D models with fine-grained details and also neural uncertainty as the information gain for the view path planning. The coarse TSDF filter viewpoints and establishes an occupancy grid map for efficient distance and occupancy query, and viewpoint filtering. The View Path Planning module first leverages volumetric representations for efficient viewpoint selection, approximates the information gain field by a MLP $\mathbf{g}_\phi$ and plans an informative view path based on A* algorithm.

B. Background of Autonomous Implicit Reconstruction

Implicit representations has shown compelling results in recent years [10], [23], [28]. The representation fitting a scene by an implicit function. This function takes the direction of a ray $\mathbf{d}$ and the location of a point $\mathbf{x}$ on the ray as inputs. Its outputs are the color value c and density ρ of the point. This function can be expressed as $\mathbf{F}_\theta(\mathbf{x}, \mathbf{d}) = (c, \rho)$ and is implemented by a MLP. To deploy the implicit representation for autonomous reconstruction, NeurAR [15] learns neural uncertainty as the information gain. It expresses

the scene function as $\mathbf{F}_\theta(\mathbf{x}, \mathbf{d}) = (\mu, \sigma, \rho)$ and yields the neural uncertainty for a viewpoint

$$\sigma_v^2 = \frac{1}{R}\sum_{r=1}^{R}\sigma_r^2 = \frac{1}{R}\sum_{r=1}^{R}\sum_{i=1}^{N} W_{ri}\sigma_{ri}^2, \tag{1}$$

where r represents a camera ray tracing through a pixel, R the number of rays sampled for a viewpoint, and N is the number of sampled point on each ray. W_{ri} is the weight of the point along the ray similar to the description in NeRF [12]. σ_{ri}^2 is the uncertainty of the color for a point learned continuously with input images added.

At each step of the autonomous reconstruction, the reconstruction module receives a set of RGBD images along with their corresponding viewpoints captured by a camera, $\{(x_i, \omega_i)\}$. With these inputs, the implicit representation $\mathbf{F}_\theta$ for the scene is updated. For the view path planning, σ_v^2 is calculated for sampled viewpoints.

C. Information Gain Field Approximation

1) Approximation $\mathbf{g}_\phi$: In (1), getting the neural uncertainty for a viewpoint is computational expensive: $R \times N$ points are passed to the MLP $\mathbf{F}_\theta$, where R scales with the image size and N scales with the scene size and the level of details needed by the reconstruction. In NeRF [12], for example, to render an image of size 800×800 given a viewpoint, R is 640,000 and N is set to 192. Though for the neural uncertainty in NeurAR, R is set to be 1000 as a coarse approximation to save computation, the number of querying $\mathbf{F}_\theta$ is still tremendous. To improve the efficiency of the information gain query, we assume the information gain for a region is generated by a smooth continuous function and propose to leverage a MLP ($\mathbf{g}_\phi$) as an implicit function approximator for the information gain field. The assumption of the smoothness is motivated by the observation that neighbouring viewpoints usually do not exhibit dramatic changes in the seen views and in their information gain. With this assumption, given a set of the information gain sampled from the field, the whole field can be interpolated.

Specifically, at each step of the planning process, given a set of sample points $\mathcal{P} = \{p_1, .., p_{N_{loc}}\}$ and its corresponding information gain $\mathcal{I} = \{I_1, ..., I_{N_{loc}}\}$, we fit the information gain distribution by learning a mapping from a point p to the information gain I

$$\mathbf{g}_\phi : p \rightarrowtail I. \tag{2}$$

The parameters ϕ is optimized by defining the L2 loss between the ground truth I and the estimation from $\mathbf{g}_\phi$, i.e. $L = \frac{1}{N_{loc}}\sum_{i=1}^{N_{loc}}(\mathbf{g}_\phi(p_i) - I_i)^2$.

With the approximation $\mathbf{g}_\phi$, getting the information gain for a viewpoint is super efficient, querying $\mathbf{g}_\phi$ for only one time compared with querying $\mathbf{F}_\theta$ for about $R \times N$ times (e.g. 100,000 times in NeuAR [15]). Also, $\mathbf{g}_\phi$ is usually much smaller than $\mathbf{F}_\theta$ in the model size and the inference is faster.

2) View Filtering with Coarse TSDF for $\mathcal{P}$: To further reduce the number of viewpoints required for the training of $\mathbf{g}_\phi$, we propose to use a coarse TSDF T to filter out viewpoints. Also, as the implicit representation $\mathbf{F}_\theta$ is not efficient in querying the status of a 3D point during the path planning, an occupancy map V is built up to accelerate the the planning.

T represents the scene using volumetric grids by integrating partial point clouds from different viewpoints using zero-crossing method [14]. From T, an occupancy map $V = V_o \cup V_e \cup V_u$ is constructed, which consists of occupied V_o, empty V_e and unobserved V_u voxels. As the volumetric maps are only for the space status query not for the fine-grained scene representation, the voxel resolution l_{res} can be very large, e.g. 10cm for a scene of size $3m \times 3m \times 3m$.

The selection for the viewpoints with directions in points $\mathcal{P}$ given current viewpoint consists of three steps: 1) sampling N_{loc} locations from the empty space V_e within a sphere centered on the current position of the camera and its radius is l_s; sampling N_{yaw} yaw and N_{pitch} pitch angles for each location; 2) choose the best three directions according to the TSDF view cost similar to that in [16]; 3) choose the best direction according to (3). As the TSDF is very coarse, the computation expense of the view cost based on it is much lower than that of (3). The information gain for these viewpoints is then normalized to [0,1] to get $\mathcal{I}$.

3) Information Gain I: In NeurAR [15], the target scene is in the center and the space for the view path planning is limited to a predefined ring area of a certain distance from the scene center. To free from the ad-hoc setting, we define information gain considering the distance of a viewpoint off the surface

$$I_v = \begin{cases} \sigma_v^2, & if\ d_{min} < d_v < d_{max} \\ e^{-\alpha|d_v - d_u|}\sigma_v^2, & others \end{cases} \tag{3}$$

where $d_{min},\ d_{max}$ are minimum, maximum depth, and $d_u = (d_{min} + d_{max})/2$. $\alpha = -2/(d_{max} - d_{min})$ is decay factor. d_v is the view depth $d_v = \frac{1}{|S_r|}\sum_{S_r} d_r$. The depth d_r of a ray $r \in S_r$ is inferred from T. S_r represent a set of rays whose depth is within the working range [d_n, d_f] of a depth camera.

D. Informative Path Planning

As the computation of the information gain for viewpoints is expensive, sampling-based methods e.g. RRT or RRT* are preferred in previous view path planning work [6], [8], [16], [20]. They typically require fewer number of sampled viewpoints, but many of them are not guaranteed to be optimal. In the following part, we propose a novel view path planning method based the A* planner, which plans shorter paths and yield better reconstruction quality.

At each step of the planning, given the current position of the camera p_s, the occupancy map V, and a set of sampled points $\mathcal{P}$ and their information gain $\mathcal{I}$, our goal is to find a local goal node and plan an informative path to obtain higher reconstruction quality with a lower path cost. For the goal node, we choose the best viewpoint among $\mathcal{P}$ by comparing

their information gain. For the informative path planning, we define a view path cost taking both the path length and the information gain of the viewpoints along the path into account,

$$IP(p) = f_d(p_s, p) - \lambda_{gain} f_d(p_s, p) G(p), \tag{4}$$

where $f_d(p_s, p)$ is the path length between a point p and the start node p_s, λ_{gain} is a gain factor and $G(p)$ represents cumulative information gained along the path. $G(p)$ is

$$G(p) = \frac{1}{N_p} \sum_{i=1}^{N_P} \mathbf{g}_\phi(p_i), \tag{5}$$

where N_P is the number of sampling points along the path between the point p and the start node p_s, p_i is the i-th point over the path. $\mathbf{g}_\phi$ is the approximator for the information gain field. For the planning with the A* algorithm, we rank the sampling priority of points to improve sampling and planning efficiency by

$$Rank(p) = IP(p) + \lambda_{rank} h_d(p, p_g), \tag{6}$$

where $h_d(p, p_g)$ is the euclidean distance between a point p and the goal node p_g and λ_{rank} is a rank factor. The step size of the planning is l_{step}.

Algorithm 1 Proposed method

Input: Images and viewpoints$\{(x_i, \omega_i)\}$, partial reconstruction 3D models $\mathbf{F}_\theta$, T, current node p_s, sampling radius r_s

Output: Updated $\{(x_i, \omega_i)\}$, $\mathbf{F}_\theta$, T, p_s

$\mathbf{F}_\theta, T \leftarrow$ Update3DScenes$(\mathbf{F}_\theta, T, \{(x_i, \omega_i)\})$
$V = V_o \cup V_e \cup V_u \leftarrow T$
$\{\omega_i\} \leftarrow$ ViewSamplingFiltering (T, V_e, r_s)
$\mathcal{P}, \mathcal{I} \leftarrow$ InformationGain$(\mathbf{F}_\theta, T, \{\omega_i\})$
$\mathbf{g}_\phi \leftarrow$ MLPFitting$(\mathcal{P}, \mathcal{I})$
$p_g \leftarrow \arg\max(\mathcal{I})$
$\{(x_i, \omega_i)\}^{\mathcal{P}} \leftarrow$ PlanExecuteViewPath$(V_e, p_s, p_g, \mathbf{g}_\phi)$
$p_s \leftarrow p_g$

IV. Results

A. Implementation details

Algorithm 1 summarizes the proposed efficient autonomous implicit reconstruction method.

1) Data: The experiments are conducted on three scenes, one small scene $cabin$ with a size of 5m × 5m × 3m, a large scene $Alexander$ with a size of 50m × 40m × 30m, and an indoor scene $childroom$ with a size of 6m × 6m × 3m. $Alexander$ is from [21], $cabin$ and $childroom$ are collected online. Similar to [15], [16], we add noise scaling approximately quadratically with depth to all rendered depth images. For $cabin$ and $childroom$, depth noise is reported from Intel Realsense L515[1] and FOV is $67.38°$. For $Alexander$, depth noise is reported from Lidar VLP16[2] and FOV is $63.5°$. The accuracy of mocap system is 0.2mm.

[1]https://www.intelrealsense.com/lidar-camera-l515/

[2]https://usermanual.wiki/Pdf/VLP16Manual.1719942037/view/

2) Implementation: Our implicit representation is implemented based on NeurAR [15] and the hyper parameters are identical. In the coarse TSDF, the near field d_n is 0.5, the voxel resolution l_{res} and the far field d_f are dependent on scenes. We set the gain factor $\lambda_d = 0.5$ in the view path cost formula (4) and the rank factor $\lambda_{rank} = 1.5$ in (6). All scene-dependent parameters are listed in Table III.

Our method runs on two RTX3090 GPUs similar to NeurAR [15]. The implicit reconstruction is on a GPU, the view path planning is on the other one. We set the maximum planned views to be 28 views for $cabin$ and $Alexander$ scenes, 40 views for $childroom$ scene.

3) Metric: We evaluate our method from two aspects including effectiveness and efficiency. For effectiveness, similar to iMAP [23] and NeurAR [15], the quality of reconstructed scenes is measured in two parts: the quality of the rendered images and the quality of the geometry of the reconstructed surface. For the quality of the rendered images, it is measured by PSNR. For $cabin$ and $Alexander$, 200 viewpoints are evenly distributed at the distances of 3m and 40m respectively away from the centers. For $childroom$, viewpoints are randomly sampled in the empty space and keep 1m away from the surface. For the geometry quality, we adopt metrics from iMAP [23]: Accuracy (cm), Completion (cm), Completion Ratio (the percentage of points in the reconstructed mesh with Completion under 1 cm for $cabin$, scene, 5 cm for $childroom$ scene, 15 cm for $Alexander$ scene). For the geometry metrics, about 300k points are sampled from the surfaces.

For efficiency, we evaluate the path length (meter) and the planning time (second). The total path length is $P.L.$. and the time is T_{GP}. For the time of our view path planning for each step of the reconstruction process, we break it into several parts for more detailed comparison: 1) the sampling time T_s to get $\mathcal{P}$ and $\mathcal{I}$, 2) the time for the $\mathbf{g}_\phi$ training T_{train}, 3) the querying time during the planning T_{query} (the number of querying points N_{query} is listed along it), 4) the pure planning time without querying $T_{planner}$. We also report the total view planning time for each step T_{SP} without the sampling time T_s, i.e. $T_{SP} = T_{train} + T_{query} + T_{planner}$. T_{train} for all the variants in Table I is about 1.3s to 1.8s. $T_{planner}$ for RRT and A* is less than 0.5s.

B. Efficacy of the Method

The efficacy of the method is evaluated regarding both the effectiveness and efficiency of our contributions, for which we design variants of our method based on implicit representations. Our method is best considering both aspects.

1) Informative Path Planning: We make **NeurAR** [15] as our baseline (V1), which uses SMC to sample views of high information gain and plans view paths by RRT. To verify the efficacy of the proposed information path planning in compared with V1, we divide it into two components: the view path cost of (4) and the graph based planning based on A*. Replacing the view path cost used to expand the tree in V1 with (4) makes V2 and replacing the RRT planner in

TABLE I: Evaluations of the effectiveness and efficiency of view path for implicit neural representation.

Variant	Filter	g_ϕ	IP	Planner	cabin PSNR↑	cabin Acc↓	cabin Comp↓	cabin C.R.↑	childroom PSNR↑	childroom Acc↓	childroom Comp↓	childroom C.R.↑	Alexander PSNR↑	Alexander Acc↓	Alexander Comp↓	Alexander C.R.↑
V1 [15]				RRT	25.69	1.56	1.16	0.68	24.92	10.94	6.33	0.66	18.15	27.11	13.51	0.62
V2			✓	RRT	26.15	1.37	1.07	0.74	26.82	9.73	5.86	0.71	21.76	21.14	12.89	0.69
V3			✓	A*	**28.67**	**1.01**	1.02	**0.76**	**28.82**	6.03	4.06	0.77	**24.57**	**15.65**	12.25	**0.72**
V4		✓	✓	RRT	26.98	1.26	1.06	0.73	26.21	8.82	5.63	0.72	21.48	21.36	13.03	0.66
V5		✓	✓	A*	28.65	1.03	**1.00**	0.77	28.02	5.53	3.92	0.76	23.88	18.41	**12.03**	**0.72**
V6(Ours full)	✓	✓	✓	A*	28.47	1.04	1.01	**0.76**	28.28	**5.44**	**3.52**	**0.78**	24.42	17.70	12.40	0.71
Variant	Filter	g_ϕ	IP	Planner	N/T_{query}	T_{SP}	T_{GP}	P.L.	N/T_{query}	T_{SP}	T_{GP}	P.L.	N/T_{query}	T_{SP}	T_{GP}	P.L.
V1 [15]				RRT	90 / 58	241	5109	44.86	104 / 1021	1723	52340	25.08	74 / 49	231	6153	430.16
V2			✓	RRT	141 / 92	92	3257	53.30	92 / 954	955	21542	25.30	119 / 75	75	2948	449.82
V3			✓	A*	222 / 144	144	4932	42.23	312 / 3489	3490	75673	**19.01**	154 / 103	103	4538	409.15
V4		✓	✓	RRT	**83 / 0.06**	1.58	1781	46.38	**224 / 0.18**	2.20	7865	26.75	165 / 0.12	2.05	1520	565.31
V5		✓	✓	A*	222 / 0.15	**1.53**	1199	**40.63**	312 / 0.22	1.96	7563	19.42	**154 / 0.10**	**1.83**	1293	411.28
V6(Ours full)	✓	✓	✓	A*	222 / 0.15	1.77	**388**	40.98	312 / 0.24	**1.94**	**1366**	19.53	154 / 0.11	1.83	**393**	**347.33**

TABLE II: Evaluations of the effectiveness and efficiency with volumetric representations.

Method	cabin PSNR↑	cabin Acc↓	cabin Comp↓	cabin C.R.↑	cabin T_{GP}	cabin P.L.	childroom PSNR↑	childroom Acc↓	childroom Comp↓	childroom C.R.↑	childroom T_{GP}	childroom P.L.	Alexander PSNR↑	Alexander Acc↓	Alexander Comp↓	Alexander C.R.↑	Alexander T_{GP}	Alexander P.L.
AEP [18]	21.45	1.52	1.87	0.35	1276	46.23	14.72	2.17	12.69	0.70	4029	29.56	16.28	20.77	22.80	0.45	1319	677
IPP [16]	18.74	1.56	2.03	0.33	447	44.27	9.02	**2.06**	67.16	0.42	1512	24.02	15.79	22.18	24.36	0.44	427	368
Our	**28.47**	**1.04**	**1.01**	**0.76**	**388**	**40.98**	**28.28**	5.44	**3.52**	**0.78**	**1366**	**19.53**	**24.42**	**17.70**	**12.40**	**0.71**	**347**	**393**

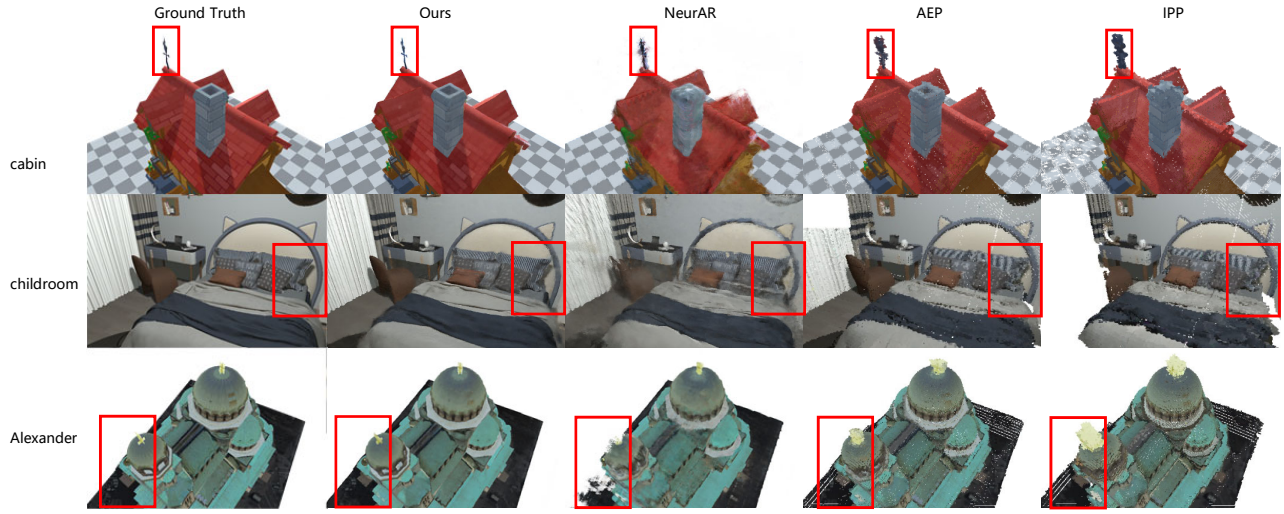

Fig. 3: Comparison of the reconstruction scenes with different methods

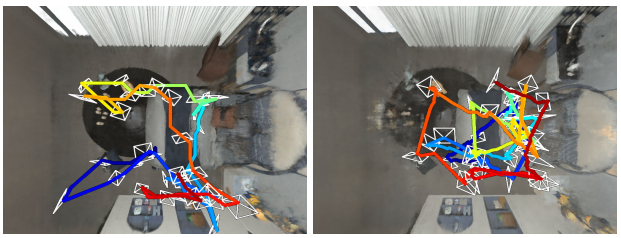

Fig. 4: Trajectories and the reconstruction results seen from top view. Left: Ours, Right: NeurAR [15]

TABLE III: Scene-dependent parameters.

Scene	l_s	l_{res}	l_{step}	d_f	N_{pitch}	N_{yaw}	d_{min}	d_{max}
cabin	3	0.1	0.2	6	3	5	2.5	4.5
childroom	1	0.1	0.2	6	5	12	1.5	3.5
Alexander	30	1	2	80	3	5	30	50

V2 with the A* planner makes V3. For these variants, the information gain for a viewpoint is computed using (1).

The metrics of V2 in Table I demonstrate the proposed view path cost alone can improve reconstruction quality than NeurAR [15]. When the view cost is combined with the A* algorithm (V3), the reconstruction quality shows a more significant improvement, which mainly attributes to the optimality of the A* algorithm. However, the planning with the A* algorithm typically requires more queries of the information gain, resulting in slower view planning. For example, for *childroom* in Table I the number of querying points for information gain N_{query} is 312 for V2 and 92 for V3; T_{SP} is 3490 seconds for V2 and 955 seconds for V3.

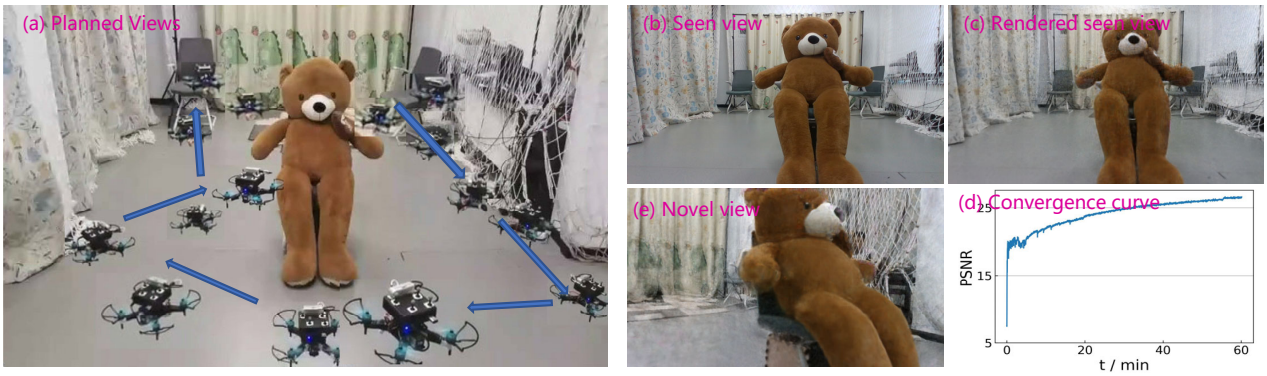

Fig. 5: The experiment on a real scene.

2) Information Gain Field Approximation: To verify the efficacy of the information gain field approximation by $\mathbf{g}_\phi$, we construct variants V4 and V5 on top of V2 and V3 respectively by querying $\mathbf{g}_\phi$ for the information gain instead of computing from (1). With the approximation, the query time of the information gain T_{query} is reduced by more than 10000 times, e.g. 3489 seconds to 0.218 seconds for 312 queries in *childroom* in Table I. The advantage of the approximation grows more significant with larger scenes or finer resolution requirement of the reconstruction. At the same time, V4 and V5 in Table I only see a very small drop in the reconstruction quality from V2 and V3, indicating the approximation of $\mathbf{g}_\phi$ is close to its ground truth.

3) Viewpoint Filtering with Coarse TSDF: To evaluate the filtering using the volumetric map, we add it to V5 and make our full method V6. With the filtering, the total view planning time is reduced by about 3 times.

C. Comparison with Volumetric Representations

Most works on autonomous reconstruction with volumetric representations are not open-source and not trivial to reimplement. We select two classic works **AEP** [18] based on OctoMap and **IPP** [16] based on TSDF, which is one of the most used representation [6], [16], [18]. We set the voxel resolution of TSDF as 1cm for *cabin* scene, 2cm for *childroom* scene, 20cm for *Alexander* scene. Table II shows that our method outperforms the reconstruction based volumetric representation in the reconstructed quality and the planning efficiency. In the childroom scene, the error measuring the accuracy of our method is higher than those of **AEP** [18] and **IPP** [16]. This is because: **AEP** [18] and **IPP** [16] do not fill the holes; our method using the implicit representation can interpolate the missing areas; the accuracy for the former only measures the non hole region but for the latter, the whole region.

Fig. 3 shows our method provides better reconstruction results. For more visual comparisons and results, we refer readers to the supplementary video. Fig. 4 demonstrates that the trajectory of our method expands in a larger region than that of NeurAR [15].

D. Ablation study

We fix the network width of $\mathbf{g}_\phi$ and use 4, 6, 8, 10 layer to study influence of the size on the reconstruction accuracy, shown in left of Fig. 6. We also change the number of views Fig. 6 to investigate the influence. The experiment is conducted on the *Alexander* scene.

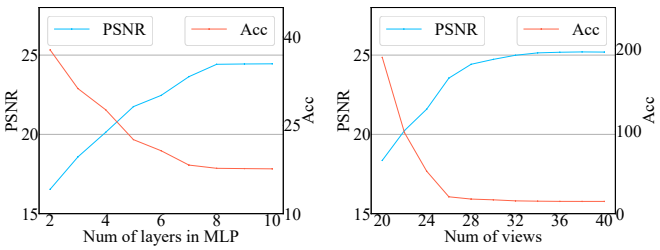

Fig. 6: Ablation study

E. Robot experiments in real scene

We deploy our proposed method on a real UAV for the reconstruction of an object placing in a room of about a size of 8m × 2.5m × 3m. The UAV is equipped with a Realsense D435i sensor[3]. The pose of the camera is provided by a Optitrack[4] system. For the scene, the UAV takes about 4 minutes to plan and executes the view path, taking 30 images. The convergence of the reconstruction module and the rendered images of the converged model are shown in in Fig. 5.

V. Conclusion

In the paper, we improve the efficiency of view path planning for autonomous implicit reconstruction by 1) approximating the information gain field by a function, and 2) combining the implicit representation with volumetric representations. Further, we propose a novel view path cost and plan view path with the A* planner. Extensive experiment shows that our method superiority in both efficiency and effectiveness.

Future directions include: overcoming the limitation of the locality of our method for large scenes, tracking the camera poses with input images instead of the optitrack system, improving the reconstructed shape of the scenes by introducing more constraints, extending the reconstruction via multi-agents.

[3]https://www.intelrealsense.com/zh-hans/depth-camera-d435i/
[4]https://optitrack.com/

REFERENCES

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[2] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon" next-best-view" planner for 3d exploration. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1462–1468. IEEE, 2016.

[3] Anjun Chen, Xiangyu Wang, Kun Shi, Shaohao Zhu, Yingfeng Chen, Bin Fang, Jiming Chen, Yuchi Huo, and Qi Ye. Immfusion: Robust mmwave-rgb fusion for 3d human body reconstruction in all weather conditions. *arXiv preprint arXiv:2210.01346*, 2022.

[4] Liang Chen, Zhitao Liu, Weijie Mao, Hongye Su, and Fulong Lin. Real-time prediction of tbm driving parameters using geological and operation data. *IEEE/ASME Transactions on Mechatronics*, 27(5):4165–4176, 2022.

[5] Guillaume Hardouin, Julien Moras, Fabio Morbidi, Julien Marzat, and El Mustapha Mouaddib. Next-best-view planning for surface reconstruction of large-scale 3d environments with multiple uavs. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1567–1574. IEEE, 2020.

[6] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484. IEEE, 2016.

[7] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1–10, 2017.

[8] Yves Kompis, Luca Bartolomei, Ruben Mascaro, Lucas Teixeira, and Margarita Chli. Informed sampling exploration path planner for 3d reconstruction of large scenes. *IEEE Robotics and Automation Letters*, 6(4):7893–7900, 2021.

[9] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floornet: A unified framework for floorplan reconstruction from 3d scans. In *Proceedings of the European conference on computer vision (ECCV)*, pages 201–217, 2018.

[10] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.

[11] Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. Taking the scenic route to 3d: Optimising reconstruction from moving cameras. In *Proceedings of the IEEEwu2014quality International Conference on Computer Vision*, pages 4677–4685, 2017.

[12] 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. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[13] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022.

[14] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011.

[15] Yunlong Ran, Jing Zeng, Shibo He, Jiming Chen, Lincheng Li, Yingfeng Chen, Gimhee Lee, and Qi Ye. Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters*, 2023.

[16] Lukas Schmid, Michael Pantic, Raghav Khanna, Lionel Ott, Roland Siegwart, and Juan Nieto. An efficient sampling-based method for online informative path planning in unknown environments. *IEEE Robotics and Automation Letters*, 5(2):1500–1507, 2020.

[17] William R Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys (CSUR)*, 35(1):64–96, 2003.

[18] Magnus Selin, Mattias Tiger, Daniel Duberg, Fredrik Heintz, and Patric Jensfelt. Efficient autonomous exploration planning of large-scale 3-d environments. *IEEE Robotics and Automation Letters*, 4(2):1699–1706, 2019.

[19] Jingjing Shen, Thomas J Cashman, Qi Ye, Tim Hutton, Toby Sharp, Federica Bogo, Andrew Fitzgibbon, and Jamie Shotton. The phong surface: Efficient 3d model fitting using lifted optimization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 687–703. Springer, 2020.

[20] Soohwan Song and Sungho Jo. Surface-based exploration for autonomous 3d modeling. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4319–4326. IEEE, 2018.

[21] Soohwan Song, Daekyum Kim, and Sunghee Choi. View path planning via online multiview stereo for 3-d modeling of large-scale structures. *IEEE Transactions on Robotics*, 38(1):372–390, 2021.

[22] Soohwan Song, Daekyum Kim, and Sungho Jo. Active 3d modeling via online multi-view stereo. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5284–5291. IEEE, 2020.

[23] 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.

[24] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022.

[25] Shihao Wu, Wei Sun, Pinxin Long, Hui Huang, Daniel Cohen-Or, Minglun Gong, Oliver Deussen, and Baoquan Chen. Quality-driven poisson-guided autoscanning. *ACM Transactions on Graphics*, 33(6), 2014.

[26] Xingbin Yang, Liyang Zhou, Hanqing Jiang, Zhongliang Tang, Yuanbo Wang, Hujun Bao, and Guofeng Zhang. Mobile3drecon: real-time monocular 3d reconstruction on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3446–3456, 2020.

[27] Boyu Zhou, Yichen Zhang, Xinyi Chen, and Shaojie Shen. Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics and Automation Letters*, 6(2):779–786, 2021.

[28] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022.