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

PredRecon: A Prediction-boosted Planning Framework for Fast and High-quality Autonomous Aerial Reconstruction

Chen Feng[2], Haojia Li[2], Fei Gao[3,4], Boyu Zhou[1,†], and Shaojie Shen[2]

Abstract— **Autonomous UAV path planning for 3D reconstruction has been actively studied in various applications for high-quality 3D models. However, most existing works have adopted *explore-then-exploit*, prior-based or exploration-based strategies, demonstrating inefficiency with repeated flight and low autonomy. In this paper, we propose PredRecon, a prediction-boosted planning framework that can autonomously generate paths for high 3D reconstruction quality. We obtain inspiration from humans can roughly infer the complete construction structure from partial observation. Hence, we devise a surface prediction module (SPM) to predict the coarse complete surfaces of the target from the current partial reconstruction. Then, the uncovered surfaces are produced by online volumetric mapping waiting for observation by UAV. Lastly, a hierarchical planner plans motions for 3D reconstruction, which sequentially finds efficient global coverage paths, plans local paths for maximizing the performance of *Multi-View Stereo* (MVS), and generates smooth trajectories for image-pose pairs acquisition. We conduct benchmarks in the realistic simulator, which validates the performance of PredRecon compared with the classical and *state-of-the-art* methods. The open-source code is released at https://github.com/HKUST-Aerial-Robotics/PredRecon.**

I. Introduction

Recently, high-quality 3D reconstruction has been an active topic in various applications including cultural relics digitalization, AR/VR, and structural inspection. Due to its high flexibility, the unmanned aerial vehicle (**UAV**) is ideal to achieve the fast, accurate, and complete 3D reconstruction of the target areas. To effectively improve reconstruction quality and efficiency, a path planning framework for autonomous aerial reconstruction is essential.

Existing reconstruction planning works [1]–[7] demonstrate unsatisfactory efficiency in reconstructing the target areas. First of all, many previous methods [1]–[4] adopt *explore-then-exploit* strategy which requires two scanning trails, or rely on coarse prior models to obtain the reconstruction paths. Such strategies present several drawbacks. 1) Two scanning trails lead to task completion inefficiency. 2) As requiring input prior model, the task cannot be fully automated. 3) They cannot guarantee accurate and complete details of the target areas owing to planning only based on coarse or prior models, which cannot adjust flight paths in *real-time* based on actual observation. Recently, online planning methods requiring a single scanning trail and not relying on prior models have been proposed [5]–[7], which partially resolve the above issues. However, the efficiency is not satisfactory enough, due to the fact that the target areas are previously unknown and significant time is distributed to explore the unknown regions. Besides, some of them demonstrate a prohibitive computation time, which usually results in undesirable stop-and-go behaviors or even requires communications with external high-end computers.

[1]School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai, China.
[2]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.
[3]State Key Laboratory of Industrail Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China.
[4]Huzhou Institute, Zhejiang University, Huzhou, China.
{cfengag, hlied, eeshaojie}@ust.hk,
fgaoaa@zju.edu.cn, zhouby23@mail.sysu.edu.cn
† **Corresponding Author**

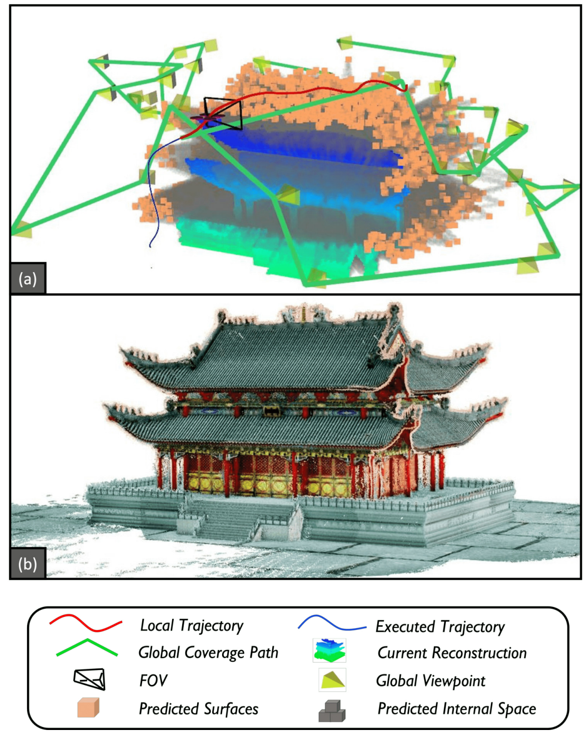

Fig. 1. (a) Illustration of the proposed framework results during executing trajectory for 3D reconstruction, (b) 3D reconstruction result of the above target produced by the proposed framework.

To address the above issues, we propose **PredRecon**, a prediction-boosted planning framework that can efficiently reconstruct high-quality 3D models for the target areas in unknown environments with a single flight. Our method is inspired by the fact that humans can reasonably infer those incomplete structures based on partial observations according to their knowledge and experience. The inferred structures or surfaces enable more purposeful viewpoints generation, which in turn allows a more efficient global coverage path of the entire target without wasting significant time on exploring unknown space. Motivated by this, we introduce a learning-based surface prediction module (**SPM**), which predicts the coarse complete surface of the target from the current partial

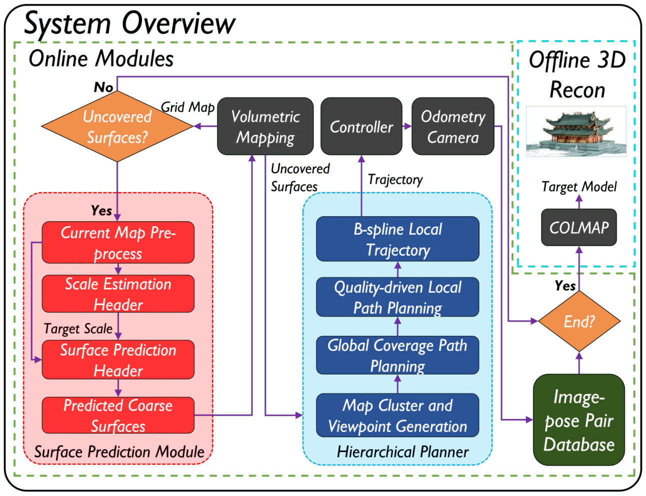

Fig. 2. The overview of the proposed prediction-boosted path planning framework for 3D reconstruction.

reconstruction. Afterwards, online volumetric mapping extracts incomplete observed surfaces from the prediction and the current reconstruction as the uncovered parts. Then, a hierarchical planner generates motions for reconstructing the uncovered surfaces in a *coarse-to-fine* manner. It first finds an efficient global path for full coverage. Secondly, a local path segment from the current pose to the next viewpoint (NBV) is generated under the guidance of the global path while optimizing the crucial factors for MVS performance. Then, the executable local trajectory is produced to acquire image-pose pairs of the target. The collected database is processed by COLMAP [8]–[10] for dense 3D reconstruction.

We compare the proposed method with the classical and *state-of-the-art* methods in a realistic simulation. Results present that our method achieves higher efficiency and better reconstruction quality in benchmark scenarios. Moreover, benchmark experiments demonstrate the higher autonomy level of our method and our method can realize *real-time* planning on typical onboard computers. The contributions of this paper are summarized as follows:

1) A surface prediction module (SPM), which directly infers the complete target surfaces from partial reconstruction information and facilitates efficient global coverage of the target without wasting significant time on extra exploration.

2) A hierarchical planner based on SPM, which sufficiently considers MVS-related factors on the fly and global coverage, achieving higher reconstruction quality and efficiency.

3) Benchmark comparisons that validate the performance of **PredRecon**. The source code of our implementation has been made public.

II. Related Work

A. Surface Prediction and Completion

Surface prediction and completion have been an essential topic in 3D reconstruction. Existing works can be roughly classified into geometry-based and learning-based methods.

The geometry-based methods predict the entire surface through geometric heuristics from partial input data. Some classical works [11]–[14] generate complete surface models using smooth interpolations from incomplete local holes. Those approaches assume that the whole surface can be inferred directly from the geometric input structure. Thus, they cannot work well during most of the flight time.

The learning-based methods take inputs from point clouds acquired through surface voxelization. They [15]–[18] directly output the complete surface model with an implicit parameterized model (deep neural network), which has better adaptiveness to complex situations. Our SPM belongs to this category. However, most existing methods suffer from unstable accuracy, primarily influenced by normalization. Hence, an extra detector is essential for predicting the scale and center of the target model. Additionally, many apply 3D CNNs for higher accuracy, while heavy architecture leads to slower inference time.

Based on this approach [15], our SPM directly uses map point cloud as input and achieves *end-to-end* surface prediction without an extra detector for normalization. Moreover, we optimize the network architecture with a more lightweight structure and more accurate performance (Sect.VI-C).

B. Path Planning for Aerial Reconstruction

For efficient and high-quality 3D reconstruction, viewpoints path planning, which selects a minimum quantity of viewpoints while maximizing contributions to reconstruction quality, has been intensely studied for years. The fundamental problem is how to model the bridge from viewpoints selection to quality. Several methods [2, 19, 20] leverage viewpoint information gain (defined as coverage of the coarse model) as the planning objectives. Furthermore, other works [1, 21] distribute a coverage hemisphere to each surface, ensuring selected viewpoints scan whole surfaces from diverse view directions.

MVS-based methods [5, 7, 22, 23] determine the optimal viewpoints considering MVS factors for better depth estimation, as this paper does. [5, 7] formulate the problem as an *information path planning problem* while [22, 23] adopt a selection strategy based on *reconstructability* heuristics. They all consider the factor of stereo matching and triangulation.

In this paper, we base our hierarchical planner on MVS-based works but with a more concise formulation of MVS heuristics cost. Moreover, it fully utilizes SPM results to generate paths with high reconstruction efficiency and quality.

III. System Overview

Fig.2 illustrates the overview of the proposed pipeline consisting of online and offline modules. The online subsystem is composed of the SPM (Sect.IV), online volumetric mapping (Sect.IV-C) and a hierarchical planner (Sect.V). SPM predicts both the scale and point cloud of the whole target model surfaces from the current partial map (Sect.IV). Then, online volumetric mapping extracts the remaining uncovered surfaces with SPM results (Sect.IV-C). After that, the hierarchical planner works to find a global path and generate a local trajectory for maximizing global coverage efficiency and MVS performance. UAV collects image-pose pairs from odometry and onboard camera (Sect.V).

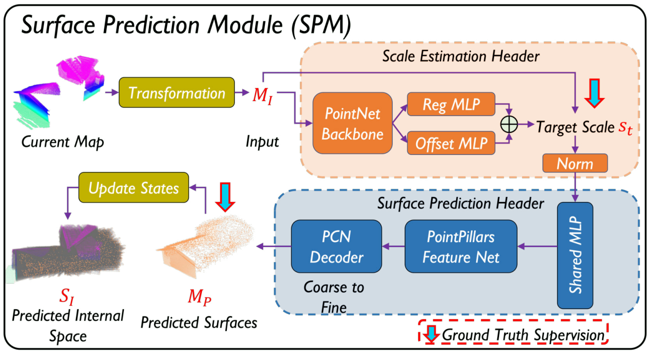

Fig. 3. The overall architecture of the proposed SPM (Sect.IV).

The online subsystem will end the flight if mapping finds no uncovered surfaces. Afterwards, the image-pose pairs database is processed using offline COLMAP to acquire the 3D reconstruction model of the target.

IV. Surface Prediction Module

SPM enables predicting the whole surfaces of the target from partial map in entirely unknown environments, as depicted in Fig.3. Surface prediction effectively decreases the redundant flight since no extra time is spent for exploring unknown environments. Moreover, it facilitates generating fewer viewpoints with the sufficient observation of the target, which reduces the complexity of the subsequent planner.

A. Data Pre-process

The input of SPM is a down-sampling point cloud M_C of the current partial map (Sect.IV-C) with the fixed quantity N_C. Different from previous works [15]–[17], we directly process each point $p_i \in M_C$ via a local transformation T_p, as follows:

$$T_p(p_i, C_C) = p_i - C_C, \tag{1}$$

where C_C is the centroid of M_C. Then, each transformed point is stored in M_I, which is sent to the prediction network.

B. Prediction Network Structure

Compared with previous point cloud completion works [15]–[17, 24], our prediction network adopts *end-to-end* manner without the extra detector for normalization. Additionally, it ensures *real-time* and lightweight requirements without 3D convolutional operation in network implementation. It consists of two headers, the scale estimation header, and the surface prediction header.

To facilitate the following surface prediction, scale estimation header is introduced to predict the coarse scale of the target. The input M_I is represented as an $N_C \times 3$ matrix containing the 3D coordinate (x, y, z) of each point. Specifically, we leverage PointNet [25] as the backbone for its permutation invariance and effective global feature extraction. Then, there are two multi-layer perceptrons (MLP) as output branches. Regression MLP directly gives a vector (x_s, y_s, z_s) indicating the scales in three axes. To further improve the scale estimation accuracy, the local feature map after PointNet is particularly processed through offset MLP to acquire corresponding offset $(\Delta x_s, \Delta y_s, \Delta z_s)$. Thus, the target scale s_t can be formulated as:

$$s_t = max(x_s + \Delta x_s, y_s + \Delta y_s, z_s + \Delta z_s). \tag{2}$$

For the training stage, we use Huber loss to supervise the scale estimations in each axis. Finally, normalization is applied on input point cloud M_I by scaling down s_t-fold.

Surface prediction header is responsible for generating the complete surfaces of the target according to the normalized M_I. We utilize a shared MLP to encode each point in the normalized M_I into the feature map F. Then, a PointPillars Feature Net [26] is performed on F as the encoder to aggregate geometric information in different areas with low computation cost for its pseudo image operation. Moreover, PointPillars is eligible for this problem since we expect the network to have the space-aware capability to extend or complete partial surfaces in different areas. Similar to PCN [15], a *coarse-to-fine* decoder is also leveraged to generate the prediction for global and local geometry learning. The fine prediction Y_{fine} and the coarse prediction Y_{coarse} both contain N_C points. For the loss function, the permutation invariant Chamfer Distance is used to supervise the difference between the network outputs with its ground truth Y_{gt}, as shown:

$$cd(X,Y) = \frac{1}{|X|}\sum_{x\in X}\underset{y\in Y}{min}||x-y||_2^2 + \frac{1}{|Y|}\sum_{y\in Y}\underset{x\in X}{min}||x-y||_2^2 \tag{3}$$

$$\mathcal{L} = cd(Y_{coarse}, Y_{gt}) + cd(Y_{fine}, Y_{gt}). \tag{4}$$

Afterwards, M_I and the inverse normalized Y_{fine} is concatenated into a $2N_C \times 3$ matrix as the predicted surfaces M_P. To determine correct viewpoints sampling space, we adopt *GHPR* [27] to process M_P to obtain the internal space S_I, which is the prohibited space for viewpoints generation.

C. Volumetric Mapping with Prediction

To online evaluate the reconstructed parts of the target, we refer to [28] to build a volumetric map, which provides partial observations for SPM. We define the surfaces that are observed from two or more different viewpoints as the complete observed surfaces. After the inference of SPM, volumetric mapping extracts those incomplete observed surfaces from the prediction as the target uncovered areas of the hierarchical planner.

V. Hierarchical Planner

With the uncovered surfaces, path planning can be formulated as generating paths to efficiently and completely cover the uncovered surfaces of the target. To realize this objective, the proposed planner takes a hierarchical planning paradigm into two steps, global coverage path planning (Sect.V-A), quality-driven local path planning for data collection and trajectory generation (Sect.V-B).

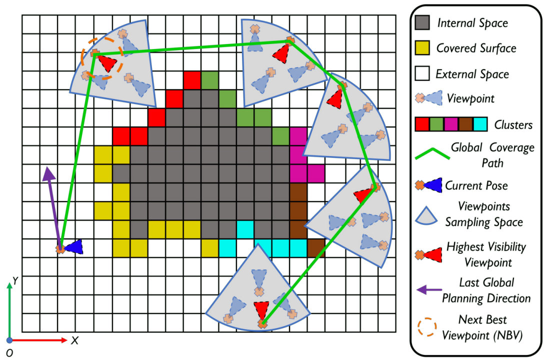

Fig. 4. Global coverage path Planning: (1) Cluster the uncovered surfaces. (2) Viewpoints generation through dual sampling. (3) The global coverage path is given by the ATSP solver. (Sect.V-A)

A. Global Coverage Path Planning

This planning stage is to output an efficient global visit sequence of the viewpoints to cover the uncovered surfaces, as illustrated in Fig.4. First of all, a clustering approach based on Euclidean distance and normal is performed on the uncovered surfaces to extract N_G clusters to be visited. Then, similar to [5], we apply the dual sampling method for the *4-DoF* viewpoints generation, which samples a set of coverage viewpoints for each cluster in their own fan-shaped cylinder from its center to normal direction, as shown in Fig.4. Lastly, we choose the viewpoint with the highest surface visibility ratio in each cluster as $V_G = \{v_g^1, v_g^2, ..., v_g^{N_G}\}$, where $v_g^i = (\mathbf{P}_g^i, \theta_g^i)$ indicating position and yaw angle. The surface visibility ratio of a viewpoint is defined as:

$$r(v, s) = \frac{\mathcal{N}(v)}{\mathcal{N}(s)}, \tag{5}$$

where v as viewpoint, s as the observed surface, $\mathcal{N}(v)$ as the number of visible points in s that can be seen from v and $\mathcal{N}(s)$ is the quantity of points in s.

To find the shortest path that passes each viewpoint from the current pose, we formulate this problem as the Asymmetric Traveling Salesman Problem (ATSP) [29]. The ATSP can be solved by existing proven algorithms through designing proper cost matrix Υ_G. Thus, we present the cost between two viewpoints $c_g(v_g^i, v_g^j)$ considers the path length and yaw change, as follows:

$$\begin{aligned} c_g(v_g^i, v_g^j) = \frac{L(\mathbf{P}_g^i, \mathbf{P}_g^j)}{v_{max}} + \\ \frac{min(||\theta_g^i - \theta_g^j||_1, 2\pi - ||\theta_g^i - \theta_g^j||_1)}{\omega}, \end{aligned} \tag{6}$$

where $L(\mathbf{P}_g^i, \mathbf{P}_g^j)$ means the path length between $\mathbf{P}_g^i$ and $\mathbf{P}_g^j$ searched by A^* algorithm in the free space, v_{max} and ω are the maximum velocity and angular change rate of yaw.

Sometimes, there exist several global coverage paths with similar cost that leads to unstable path optimization results, which introduces inconsistent flight directions and low efficiency. Accordingly, global consistency should be essentially taken into account to generate stable solutions. We define the last global planning direction (a vector from last current position $\mathbf{P}_{cur}^{last}$ to last NBV $\mathbf{P}_{nbv}^{last}$) d_g^{last}, and introduce glocal consistency cost $c_{GC}(v_g^i)$ by:

$$d_g^{last} = \frac{\mathbf{P}_{nbv}^{last} - \mathbf{P}_{cur}^{last}}{||\mathbf{P}_{nbv}^{last} - \mathbf{P}_{cur}^{last}||_2}, \tag{7}$$

$$c_{GC}(v_g^i) = arccos \frac{\mathbf{P}_g^i - \mathbf{P}_{cur}^{now}}{||\mathbf{P}_g^i - \mathbf{P}_{cur}^{now}||_2} \cdot d_g^{last}. \tag{8}$$

Then, we can give the complete form of Υ_G with the viewpoints index set $\zeta = \{1, 2, ..., N_G\}$ as:

$$\Upsilon_G(k, h) = \begin{cases} 0, & k == h \, or \, h = 0 \\ c_g(v_g^k, v_g^h), & k, h \in \zeta \\ [\beta_1 c_g(v_g^k, v_g^h) + & k == 0 \, and \, h \in \zeta \\ \beta_2 c_{GC}(v_g^h)], & \end{cases} \tag{9}$$

Therefore, through solving the above ATSP with Υ_G, we can find the efficient global coverage path starting from the current pose to visit the whole uncovered surfaces.

B. Quality-driven Local Path Planning

Global planning mainly focuses on fast and complete coverage of the target. To further improve the reconstruction quality, local planning optimizes a segment path from the current pose to NBV, which fully considers MVS-related factors, as depicted in Fig.5.

Different from global planning, the cluster covered by the local segment is further subdivided into smaller clusters while viewpoints sampling space in local planning is determined by two neighboring clusters, as shown in Fig.5. Local viewpoints set is represented as the form of $V_L = \{VP_1 : \{v_l^{1,1}, v_l^{1,2}, ..., v_l^{1,n}\}, ..., VP_i : \{v_l^{i,1}, v_l^{i,2}, ..., v_l^{i,k}, ...\}\}$, and clusters shown as $\mathcal{C}_L = \{cls_1, cls_2, ..., cls_j, ...\}$.

Many previous studies [9, 30, 31] demonstrated the high-quality MVS reconstruction thoroughly depending on the following factors, including visibility $\mathcal{S}_{vis}$, relative distance $\mathcal{S}_{dis}$ and triangulation angle $\mathcal{S}_{ang}$, presented in Eq.10, 11, 12, 13. To optimize MVS performance of a local path, we decompose the MVS structure into several basic triangulation units, which is defined as each of two neighboring viewpoints in the local path with their co-visible cluster surface. Furthermore, the MVS performance of this path can be viewed as the reconstruction quality Q sum of all triangulation units in this path. Then, Q of a triangulation unit can be written as:

$$Q(v_1, v_2, s) = \mathcal{S}_{vis} \cdot \mathcal{S}_{dis} \cdot \mathcal{S}_{ang}, \tag{10}$$

where the cluster surface s under two viewpoints v_1 and v_2.

$\mathcal{S}_{vis}$ is the score for the visibility ratio ($r \in [0, 1]$) of two viewpoints, shown as:

$$\mathcal{S}_{vis}(v_1, v_2, s) = \frac{r(v_1, s) + r(v_1, s)}{2}. \tag{11}$$

Let dis_1 and dis_2 be the distances from two viewpoints to the surface centroid. We expect $\mathcal{S}_{dis}$ to be close to 1 which leads to similar resolution in two viewpoints images for better depth estimation. The formula follows:

$$\mathcal{S}_{dis}(v_1, v_2, s) = \frac{min(dis_1, dis_2)}{max(dis_1, dis_2)}. \tag{12}$$

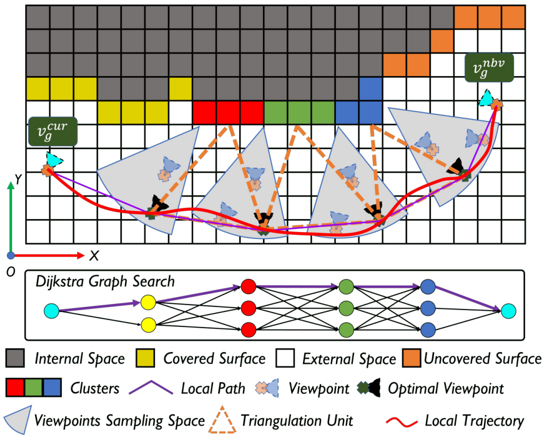

Fig. 5. Quality-driven local path planning based on the graph search. Through fully considering MVS-related factors, a reconstruction quality-driven local path is produced with its corresponding trajectory. (Sect.V-B)

$\mathcal{S}_{ang}$ measures the triangulation performance, both accuracy and matchability. Let ϵ be the angle between vec_1 and vec_2. ε_1 is the angle between the normal $\mathcal{N}_s$ of s and vec_1 while ε_2 is the same for vec_2. Hence, $\mathcal{S}_{ang}$ can be written as:

$$\begin{aligned} vec_h &= \mathcal{C}_s - v_h, \\ \mathcal{S}_{ang}(v_1, v_2, s) &= exp(-(\frac{\epsilon - \epsilon_d + \varepsilon_1 - \varepsilon_2}{\kappa})^2), \end{aligned} \tag{13}$$

where $\mathcal{C}_s$ is the centroid of s, ϵ_d is the desired triangulation angle and κ is a small constant value for numerical stability.

Thus, we can formulate the MVS heuristics cost c_{MVS} and total cost c_l with movement cost as:

$$c_{MVS}(v_1, v_2, s) = \frac{1}{Q(v_1, v_2, s)}, \tag{14}$$

$$c_l(v_1, v_2, s) = \alpha_1 c_{MVS}(v_1, v_2, s) + (1 - \alpha_1) c_g(v_1, v_2). \tag{15}$$

Assuming there are N_L clusters totally, the number of V_L should be $N_L + 1$ to satisfy the N_L defined triangulation units. To optimize the quality-driven cost c_l of the local path, we formulate it as a graph search problem. Then, the Dijkstra algorithm is adopted to search for the optimal local path, $\mathcal{P}_L = \{v_l^{1,i_1}, v_l^{2,i_2}, ..., v_l^{N_L+1, i_{N_L+1}}\}$ that minimizes the proposed cost:

$$min \sum_{k=1}^{N_L} c_l(v_l^{k,i_k}, v_l^{k+1,i_{k+1}}, cls_k). \tag{16}$$

Lastly, through leveraging [32], we convert the local path $\mathcal{P}_L$ to the safe, smooth, dynamically feasible, and minimum-time B-spline local trajectory considering MVS performance to realize an effective collection of image-pose pairs.

VI. Experiments

A. Implementation Details

To train our SPM, we use a synthetic CAD model set, Houses3K [33] to create a construction scene dataset containing partial and complete point clouds. Also, we collect other types of construction models in Unreal Engine (UE4[1]). Specially, we leverage Blender[2] to generate partial point clouds with 12900 models from different construction categories. Additionally, we set $N_C = 8192$ in the data pre-processing phase. As for training details, the SPM is trained for 200 epochs on single NVIDIA RTX 3070Ti taking 13 hours. We choose the Adam [34] optimizer during training with an initial learning rate of 1e-4 with a batch size of 16, decaying to 1e-5 at 150 epochs.

In hierarchical planning, we set $\beta_1 = 1.0$ and $\beta_2 = 5.0$ in Eq.9, $\epsilon_d = 22.5°$ and $\kappa = 0.2$ in Eq.13, and $\alpha_1 = 0.8$ in Eq.15. In global coverage path planning, the ATSP is solved through a Lin-Kernighan-Helsgaun heuristic solver [35].

In all experiments, a geometric controller [36] is used for tracking control of the (x, y, z, θ) trajectory. SPM runs on an NVIDIA RTX 3070 Ti (GPU Memory-Usage: ~**1GB**) and other modules run on an Intel Core i9-10900K CPU.

TABLE I

Path Planning and 3D Reconstruction results in two scenarios.

	Method	Prior Model	Path Length (m)	Time (s)	Recall (%)	Precision (%)	F-score (%)
Palace	Plan3D [2]	✗	375.5	507.7	74.48	82.57	78.32
	CAPP [1]	✓	243.6	322.6	69.21	85.86	76.64
	FUEL [6]	✗	371.1	469.8	40.31	38.38	39.32
	Ours	✗	**213.1**	**252.7**	**74.67**	**86.45**	**80.13**
Village House	Plan3D [2]	✗	239.3	310.6	64.28	72.86	68.30
	CAPP [1]	✓	193.4	242.3	80.30	**84.60**	82.40
	FUEL [6]	✗	405.1	506.8	44.35	36.46	40.02
	Ours	✗	**153.2**	**184.6**	**84.54**	83.13	**83.83**

B. Benchmark Comparisons

We conduct simulation Experiments in a realistic simulator, AirSim in UE4. We benchmark it in two highly textured scenarios, ***Palace*** ($15 \times 25 \times 14m^3$) and ***Village House*** ($14 \times 11 \times 12m^3$). The proposed method is compared with three methods: Plan3D [2] (*explore-then-exploit*), CAPP [1] (prior-based) and FUEL [6] (exploration-based). There is no open source code for Plan3D [2] and CAPP [1], so we use our implementation. A UAV mounting a forward-looking camera with FOV $[80°, 60°]$ is adopted as the experimental platform. It captures images with a resolution 1280×720 px. In both scenarios, we limit the $v_{max} = 0.85m/s$ and $\omega = 0.5rad/s$. Plan3D [2] firstly executes a pre-defined flight for the coarse model, and then generates the global path using our planner. CAPP [1] produces a global coverage path also by our planner according to input prior model. As for FUEL [6], it collects image-pose pairs of the target while exploring the unknown environments containing the target. The collected data of each method is processed through COLMAP to obtain reconstructed 3D models.

We evaluate their performance by two metrics, efficiency (path length and time) and reconstruction quality (*F-score*). The average comparison results are listed in Table.I and Fig.6. Compared with the other methods, we both achieve much shorter time and path length, primarily since our planner gives a more efficient global coverage path with the

[1]https://www.unrealengine.com/en-US/
[2]https://www.blender.org/

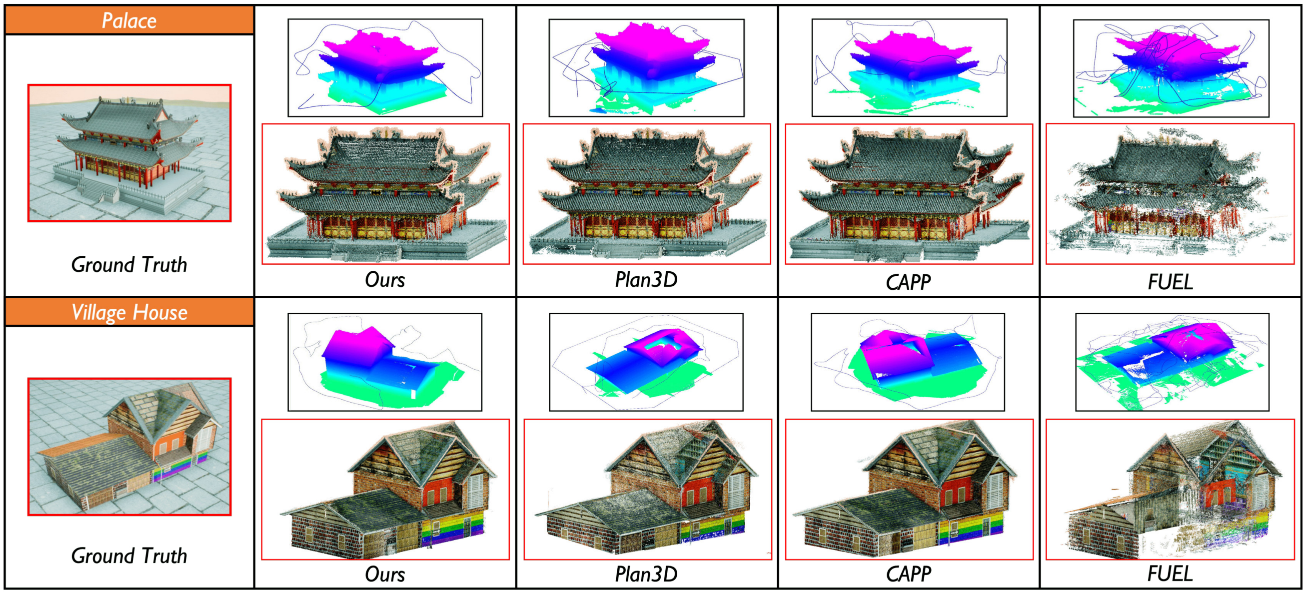

Fig. 6. Benchmark comparisons (Reconstructed 3D models and volumetric maps with the executed trajectories) of the proposed method, Plan3D [2], CAPP [1] and FUEL [6] in two scenarios (***Palace*** and ***Village House***).

TABLE II
COMPUTATION TIME OF EACH MODULE.

	SPM	Global Planning	Local Planning	Traj. Opt.	Total Comp.
Time (ms)	∼26.8	∼93.5	∼0.5	∼3.7	∼124.7

TABLE III
POINT CLOUD COMPLETION PERFORMANCE COMPARISONS.

Method	#Param(M)	L1_CD (1e-3m)	L2_CD (1e-4m)	F-score (%)
our SPM	**28.20**	**13.6404 / 9.4461**	**14.7100 / 3.9368**	**52.6050 / 68.6693**
PCN [15]	28.91	15.5221 / 10.4897	18.3987 / 4.7431	50.1210 / 65.7207

support of SPM predictions. As for reconstruction quality, we refer to the evaluation process and metrics in [37]. First, we perform point cloud alignment between the reconstructed model and ground truth. Then, two point clouds are uniformly resampled with a voxel size of $0.05m$, which are compared by $Precision$ and $Recall$. $Precision$ is presented as the percentage of reconstructed points close to a ground truth point while $Recall$ is defined as the percentage of ground truth points close to a reconstructed point. We set the distance between two points is less than $0.1m$, which are close points. Afterwards, the *F-score* is formulated as $F-score = \frac{2(Precision \times Recall)}{Precision+Recall}$. Fig.6 and Table.I depicts the reconstruction quality in two scenarios of each reconstructed model by four methods. Obviously, the proposed method achieves higher $Precision$, $Recall$, and *F-score*, mainly because our local planning aims to optimize MVS performance, and our method real-time replans the paths for complete details whenever predictions and map are updated. Although our $Precision$ is slightly lower than CAPP [1] in ***Village House*** scenario, no prior model is required in our method.

As shown in Table.II, the proposed system can finish planning once in approximately $100ms$, which enables enough frequency for *real-time* planning on the onboard computer of a realistic UAV.

C. SPM Prediction Performance

Compared with the point cloud completion task, the surface prediction in our system is more difficult since no exact scale and center are given for normalization. However, under Chamfer Distance and *F-score* metrics, our SPM without prior scale and center still outperforms PCN [15] in the above task using the generated data (Sect.VI-A) (Left) and ShapeNet dataset (Right), as listed in Table.III. Considering reconstructed surfaces, PCN [15] produces smoother surfaces than coarse prediction results generated by SPM.

VII. CONCLUSIONS

In this paper, we propose a prediction-boosted planning framework for efficient high-quality 3D reconstruction with an autonomous single trail. The proposed SPM predicts complete surfaces from the partial map to provide global information for the path planner. Based on the SPM, a hierarchical planner sequentially plans motions for 3D reconstruction. It finds efficient global coverage paths, optimizes reconstruction quality-driven local paths to improve MVS performance, and generates smooth corresponding local trajectories. The method significantly improves reconstruction efficiency and quality via introducing SPM and considering MVS-related factors. Challenging benchmark in realistic simulation shows the competence of **PredRecon** compared with the existing classical and *state-of-the-art* methods.

The limitation of our method is insufficient real-world tests as well as the limited generalizability and robustness of SPM. In the future, we plan to further optimize SPM architecture for better data representation and implement more challenging real-world tests.

References

[1] H. Zhang, Y. Yao, K. Xie, C.-W. Fu, H. Zhang, and H. Huang, "Continuous aerial path planning for 3d urban scene reconstruction." *ACM Trans. Graph.*, vol. 40, no. 6, pp. 225–1, 2021.

[2] B. Hepp, M. Nießner, and O. Hilliges, "Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 1, pp. 1–17, 2018.

[3] Q. Kuang, J. Wu, J. Pan, and B. Zhou, "Real-time uav path planning for autonomous urban scene reconstruction," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1156–1162.

[4] X. Zhou, K. Xie, K. Huang, Y. Liu, Y. Zhou, M. Gong, and H. Huang, "Offsite aerial path planning for efficient urban scene reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–16, 2020.

[5] S. Song, D. Kim, and S. Choi, "View path planning via online multiview stereo for 3-d modeling of large-scale structures," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 372–390, 2021.

[6] 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.

[7] S. Song, D. Kim, and S. Jo, "Active 3d modeling via online multi-view stereo," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5284–5291.

[8] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[10] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, "A vote-and-verify strategy for fast spatial verification in image retrieval," in *Asian Conference on Computer Vision (ACCV)*, 2016.

[11] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, pp. 1–13, 2013.

[12] J. Davis, S. R. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *Proceedings. First international symposium on 3d data processing visualization and transmission*. IEEE, 2002, pp. 428–441.

[13] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, and C. Silva, "State of the art in surface reconstruction from point clouds," *Eurographics 2014-State of the Art Reports*, vol. 1, no. 1, pp. 161–185, 2014.

[14] W. Zhao, S. Gao, and H. Lin, "A robust hole-filling algorithm for triangular mesh," *The Visual Computer*, vol. 23, no. 12, pp. 987–997, 2007.

[15] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 728–737.

[16] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "Grnet: Gridding residual network for dense point cloud completion," in *European Conference on Computer Vision*. Springer, 2020, pp. 365–381.

[17] L. Pan, X. Chen, Z. Cai, J. Zhang, H. Zhao, S. Yi, and Z. Liu, "Variational relational point completion network," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8524–8533.

[18] J. Shi, L. Xu, P. Li, X. Chen, and S. Shen, "Temporal point cloud completion with pose disturbance," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4165–4172, 2022.

[19] A. Hornung, B. Zeng, and L. Kobbelt, "Image selection for improved multi-view stereo," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[20] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, "Automatic view selection using viewpoint entropy and its application to image-based modelling," in *Computer Graphics Forum*, vol. 22, no. 4. Wiley Online Library, 2003, pp. 689–700.

[21] M. Roberts, D. Dey, A. Truong, S. Sinha, S. Shah, A. Kapoor, P. Hanrahan, and N. Joshi, "Submodular trajectory optimization for aerial 3d scanning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5324–5333.

[22] N. Smith, N. Moehrle, M. Goesele, and W. Heidrich, "Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark," 2018.

[23] C. Peng and V. Isler, "Adaptive view planning for aerial 3d reconstruction," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2981–2987.

[24] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "Pf-net: Point fractal network for 3d point cloud completion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7662–7670.

[25] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[26] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[27] S. Katz and A. Tal, "On the visibility of point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1350–1358.

[28] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," *arXiv preprint arXiv:1903.02144*, 2019.

[29] Z. Meng, H. Qin, Z. Chen, X. Chen, H. Sun, F. Lin, and M. H. Ang, "A two-stage optimized next-view planning framework for 3-d unknown environment exploration, and structural reconstruction," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1680–1687, 2017.

[30] O. Mendes, S. Hadfield, N. Pugeault, and R. Bowden, "Next-best stereo: Extending next-best view optimisation for collaborative sensors," 2016.

[31] O. Mendez, S. Hadfield, N. Pugeault, and R. Bowden, "Taking the scenic route to 3d: Optimising reconstruction from moving cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4677–4685.

[32] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[33] D. Peralta, J. Casimiro, A. M. Nilles, J. A. Aguilar, R. Atienza, and R. Cajote, "Next-best view policy for 3d reconstruction," *arXiv preprint arXiv:2008.12664*, 2020.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] K. Helsgaun, "An effective implementation of the lin–kernighan traveling salesman heuristic," *European journal of operational research*, vol. 126, no. 1, pp. 106–130, 2000.

[36] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 5420–5425.

[37] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.