2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Active 3D Modeling via Online Multi-View Stereo

Soohwan Song, Daekyum Kim, and Sungho Jo

***Abstract*— Multi-view stereo (MVS) algorithms have been commonly used to model large-scale structures. When processing MVS, image acquisition is an important issue because its reconstruction quality depends heavily on the acquired images. Recently, an explore-then-exploit strategy has been used to acquire images for MVS. This method first constructs a coarse model by exploring an entire scene using a pre-allocated camera trajectory. Then, it rescans the unreconstructed regions from the coarse model. However, this strategy is inefficient because of the frequent overlap of the initial and rescanning trajectories. Furthermore, given the complete coverage of images, MVS algorithms do not guarantee an accurate reconstruction result.**

In this study, we propose a novel view path-planning method based on an online MVS system. This method aims to incrementally construct the target three-dimensional (3D) model in real time. View paths are continually planned based on online feedbacks from the partially constructed model. The obtained paths fully cover low-quality surfaces while maximizing the reconstruction performance of MVS. Experimental results demonstrate that the proposed method can construct high quality 3D models with one exploration trial, without any rescanning trial as in the explore-then-exploit method.

I. Introduction

Three-dimensional (3D) modeling of large-scale structures is an important and ongoing research issue [1]. To address this issue, multi-view stereo (MVS) algorithms [2] [3] are widely used because they can estimate wider depth ranges than stereo cameras or RGB-D sensors. MVS is an offline method that processes a collection of calibrated images in a batch to reconstruct the corresponding 3D model. The reconstruction quality of MVS greatly depends on the collected images [4] [5]. Therefore, it is important to acquire a set of images that can fully cover a target structure to maximize the reconstruction performance of MVS.

In order to determine an optimal trajectory that generates the best 3D model in MVS, view path-planning algorithms are widely used. As a view path-planning method, an inspection-planning approach [6] [7] [8] is commonly employed when reconstructing a large-scale structure. Based on the assumption that a prior model of the target structure is known, the inspection approach aims to compute camera trajectories that generates the complete surface coverage of the prior model. However, for most real-world cases, prior information of the target structure is unavailable. To address this issue, studies have proposed the *explore-then-exploit* method [1] [5] [9] [10]. This method first constructs a coarse model by exploring an entire scene using a fixed trajectory within a safe area. Then, based on the coarse model, it plans an inspection path for the entire model.

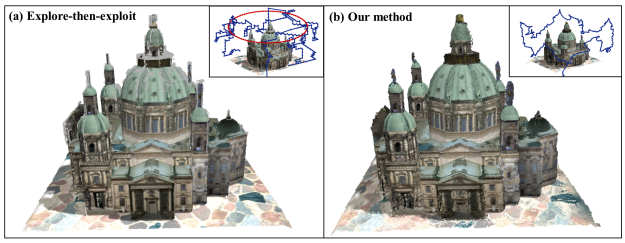

Fig. 1. 3D reconstruction results. (a) The explore-then-exploit method [9] with offline MVS [2]. It first constructs an initial coarse model from a default trajectory (in red) and then computes the coverage path (in blue). It takes about 11 hours in total (3 hours for coarse model and 8 hours for detail reconstruction) to process all acquired images. (b) Our method: view-planning and online MVS system. It reconstructs a 3D scene in real-time and completes the modeling process in one exploration trial.

However, modeling performance of the explore-then-exploit method can be degraded for several reasons. First, this method sometimes generates overlapped trajectories, which leads to inefficient performance in time because the camera needs to scan the same areas repeatedly. Second, even though given images fully cover the target structure, the MVS algorithms are not guaranteed to generate a complete and accurate reconstructed model due to textureless scenes, short baseline distances, and occlusions. Third, the MVS algorithms usually take a long time to process the images, which makes the entire modeling process extremely slow.

To address these problems, we propose a novel view path-planning method based on an online MVS system. The proposed online MVS system extends monocular mapping algorithms [11] [12] [13], which focus only on the local dense mapping, to be adaptable to constructing a large-scale model. The proposed system handles large amounts of noises and outliers in 3D data using several post-processing steps, including noise filtering and depth interpolation. Unlike the offline methods, our method incrementally constructs the large-scale 3D models using the surfel mapping method [14] in real-time. By doing so, it enables to evaluate the completeness of a model by analyzing the quality of the reconstructed surface. The proposed method iteratively plans view paths using the reconstruction quality feedback. It determines the best views to acquire reference and source images using the heuristic information of MVS. Based on the determined views, it provides an optimal camera trajectory that satisfies the followings: i) to cover low-quality surfaces

*This work was supported by the National Research Foundation of Korea funded by the Ministry of Education (No.2016R1D1A1B01013573) and by the Technology Innovation Program funded by the Ministry of Trade, industry & Energy (MI, Korea) (No.10070171).

Soohwan Song, Daekyum Kim, and Sungho Jo are with School of Computing, KAIST, Daejeon 34141, Republic of Korea. {dramanet30, daekyum, shjo}@kaist.ac.kr

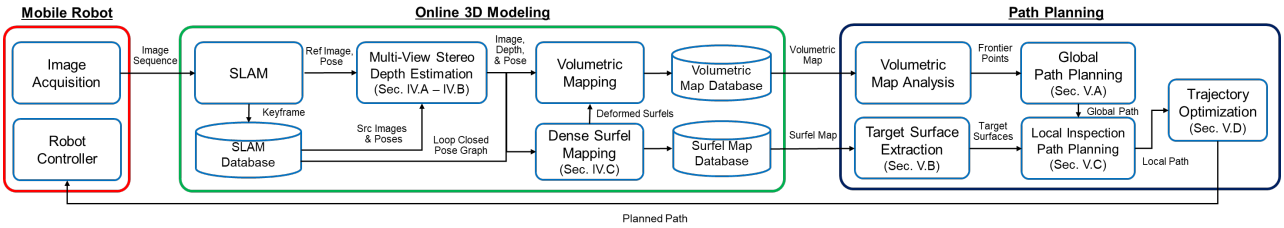

Fig. 2. Overall system architecture of the proposed active 3D modeling framework (see Section III for more details).

in the current scene, and ii) to improve the stereo matching performance. This planning method then constructs a complete and accurate 3D model in one exploration trial, without any rescanning trial like the explore-then-exploit (Fig. 1).

The contributions of this paper are summarized as follows: (i) Unlike existing approaches, we employ an online MVS system based on the monocular mapping algorithm for constructing 3D models to address the view path-planning problem. (ii) We propose a view path-planning method that performs a trajectory optimization and view selection to maximize the performance of MVS reconstruction. (iii) We empirically evaluate the proposed method in two simulated scenarios using a Micro Aerial Vehicle (MAV). The effectiveness and applicability of the proposed method are evaluated in comparison with existing methods.

II. Related Work

The problem of computing optimal views during the construction of a 3D model is known as view planning or active vision [15] [16] [17]. Different types of view-planning methods have been developed and used for different reconstruction algorithms. Real-time mapping algorithms [18] [19] accumulate 3D data directly from a RGB-D sensor. Because these algorithms can evaluate modeling completeness online, exploration approaches are frequently used to plan view paths by simultaneously identifying uncovered areas from the partially-reconstructed model. Most approaches iteratively determined the *next-best-view* (NBV) [20] [21] or view path [22] for exploration planning by analyzing frontiers in a volumetric map. Song and Jo [23] proposed a surface-based exploration method, which simultaneously considers unexplored regions and reconstruction quality. Our work extends this method [23] by incorporating with the trajectory optimization step to improve the performance of MVS.

Several methods [24] [25] have been proposed to model the target scene directly from a sparse point cloud using structure-from-motion (SfM). Hoppe et al. [24] proposed an online SfM framework that provides a visual feedback of reconstruction quality to a human operator. Haner and Heyden [25] proposed a covariance propagation method that determines a view sequence to minimize the reconstruction uncertainty of the sequential SfM.

The MVS algorithms have been used in large-scale structure modeling because they can estimate a wide depth range of the target scene. Some approaches [26] [27] acquired images for MVS reconstruction using a simple lawn mower or circular trajectories in a safe overhead area. However, these approaches do not guarantee complete coverage of the target structure. Therefore, an explore-then-exploit approach is generally adopted in 3D modeling systems based on an MVS algorithm [28] [9] [5] [10]. Roberts et al. [9] proposed the modeling of surface coverage by observing a hemispheric area around the surface. This method computes a coverage trajectory, such that the camera observes surfaces from diverse viewpoints by maximizing the observing area of the hemispheres. Huang et al. [5] proposed a relatively fast MVS algorithm that reconstructs coarse 3D models. Their method iteratively determines the NBVs by analyzing the coverage of a tentative surface model.

There have been only a few methods [29] [30] [31] that deal with an online MVS algorithm. Mendez et al. [29] [30] used an online dense stereo matching algorithm [32] based on deep learning. They proposed a next-best-stereo method that determines the best stereo pair to maximize stereo matching performance. Forster et al. [31] addressed a view-planning problem that acquires an informative motion trajectory for monocular dense depth estimation. These approaches concentrated only on the local depth estimation, whereas global modeling was not considered. On the other hand, the proposed method deals with not only the local trajectory for local mapping, but also the global trajectory for constructing a global model.

III. System Overview

Fig. 2 illustrates an overall system architecture of the proposed framework, which is composed of two modules: *online 3D modeling* and *path planning*. The 3D modeling module constructs 3D models online from an obtained image sequence. It first computes the camera poses from a SLAM system [33] and generates a depth map of a scene using a monocular mapping method [11]. The obtained depth maps are integrated into a volumetric map $\mathcal{M}$ and a surface model $\mathcal{F}$ simultaneously. The volumetric map [19] categorizes an environment as three states (unknown, free, and occupied), which is required for planning a collision-free path. The surface model is represented as a collection of dense point primitives, surfels, which is efficient for surface deformation and rendering.

The path-planning module computes the exploration paths to reconstruct a target structure. The module first plans a global path by analyzing the volumetric map in order to explore a large unknown area. Then, the module computes a

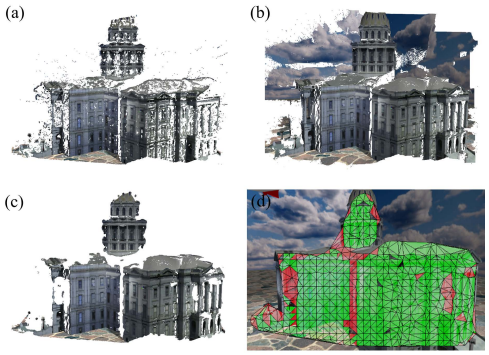

Fig. 3. Examples of a reconstructed point cloud of a scene by (a) initial depth, (b) fully-interpolated depth, and (c) partially-interpolated depth. (d) Delaunay triangulation result of a set of support depth points. The weighted least squares (WLS) filter is applied in planar areas (green triangles), excluding non-planar areas (red triangles).

local inspection path that covers the defectively reconstructed surfaces in the surface model. The planned local path is refined to maximize the performance of MVS reconstruction. The mobile robot constantly navigates along the planned path while constructing the 3D models of the target structure.

IV. Online Multi-View Stereo

The online MVS system is based on a monocular dense mapping algorithm [11]. The system obtains images and the corresponding camera poses using a SLAM module. We use a keyframe-based SLAM method [33] that computes a camera pose by estimating the sparse map points from selected keyframes. When a new keyframe is extracted, the image-pose pair is stored in a database, and the keyframe is set as a reference image for depth estimation. Unlike existing monocular mapping algorithms [11] [12], which use subsequent sequential images as source images for stereo matching, our method selects the best set of source images in an online and active manner to improve the depth estimation performance. (see Section V.E for active image selection)

A. Depth Estimation

A depth map D_t of a reference image I_{ref} is obtained by pixel-wise stereo matching, given a source image I_t at time t. Obtained depth maps $D_1, ..., D_T$ are sequentially integrated using the *REMODE* [11], a recursive Bayesian estimation approach, which integrates sequential depths. For each pixel, the algorithm recursively updates the mean depth d and its variance σ^2, which follow a Gaussian distribution, and the inlier probability ρ. A depth estimate is considered converged when $\rho > \theta_{inlier}$ and $\sigma^2 < \theta_{var}$. The algorithm then applies a regularization filter, a variant of the total variation.

B. Depth Post-Processing

The depth maps estimated from REMODE relatively have more number of unconverged pixels (in textureless regions) and outliers in converged pixels than those from the offline MVS methods [2] [3]. Therefore, we first remove the outlier depths by checking whether a depth point is supported by neighboring pixels as proposed in [34]. Next, we interpolate the unconverged depth values while enforcing depth smoothness of the coplanar surfaces. We employ the fast weighted least squares (WLS) approach [35] for depth interpolation. WLS performs the image-guided, edge-preserving interpolation by solving a global optimization problem.

Although the WLS method is usually effective in textureless or planar regions, it causes noise in regions near depth discontinuities and non-planar surfaces [36]. Furthermore, depth values could be interpolated into empty areas, such as the sky or distant blurry regions (Fig. 3b). To address this issue, we extract piecewise planar areas in a target structure and apply the WLS filter only to the planar areas.

First, we divide the entire image region into rectangular grids and determine a sparse set of support depth points by selecting the median depth point within each grid. Then, we compute a two-dimensional (2D) Delaunay triangulation of the pixel locations of the support depth points using the fast triangulation method [37]. Triangular meshes each of whose edges are longer than a certain threshold in 3D are eliminated from the triangular set. We segment an image into a set of triangular regions $\mathcal{T} = [T_1, ..., T_K]$ according to the constructed triangular meshes. Each triangular region T_k can be described by its plane parameters $\boldsymbol{\tau}_k = (\tau_k^1, \tau_k^2, \tau_k^3) \in R^3$; a depth in a pixel p is defined as $d_p = \tau_k^1 p_x + \tau_k^2 p_y + \tau_k^3$ where p_x and p_y denote p's x- and y-coordinates [38]. Given a depth map D' interpolated by the WLS filter, we define the planarity of each triangular region T_k as

$$f_{pl}(T_k) = \frac{1}{\#(T_k)} \sum_{p \in T_k} \mathbf{1}[|\tau_k^1 p_x + \tau_k^2 p_y + \tau_k^3 - d'_p| < d_{thr}] \tag{1}$$

where $\#(T_k)$ is the number of pixels in T_k, d_{thr} is the depth threshold, and $\mathbf{1}[\cdot]$ is the indicator function. This measure simply represents the ratio of interpolated depths that are consistent with a plane parameter $\boldsymbol{\tau}_k$. If the ratio is higher than a threshold (0.95 in this study), the region is labeled as a planar area (Fig. 3d). Similar to the method in [36], the interpolated depth d' is applied in the planar area, and the initial depth d is used in the non-planar area. Fig. 3c illustrates an example of the post-processed result.

C. Surfel Mapping

The surface model $\mathcal{F}$, which represents a reconstructed surface, is composed of 3D surfels, where each surfel has the following attributes: a 3D position, normal, color, weight, and radius. We employ the surfel initialization and surfel fusion method as described in [14]. An initial weight w is directly mapped by a depth variance σ^2 as: $w = \sigma_{max} - \sqrt{\sigma^2}$, where σ_{max} is user defined maximum standard deviation (1.0 in this study). Weight of an interpolated depth is set as a constant $w_{const} = \sigma_{max} - \sqrt{\theta_{var}}$. As in [14], we label the surfels that have not been fused in a period as inactive, and filter out low-weight inactive surfels.

D. Processing Loop Closing

When the SLAM module performs loop closing, our method deforms the surface model and the volumetric map

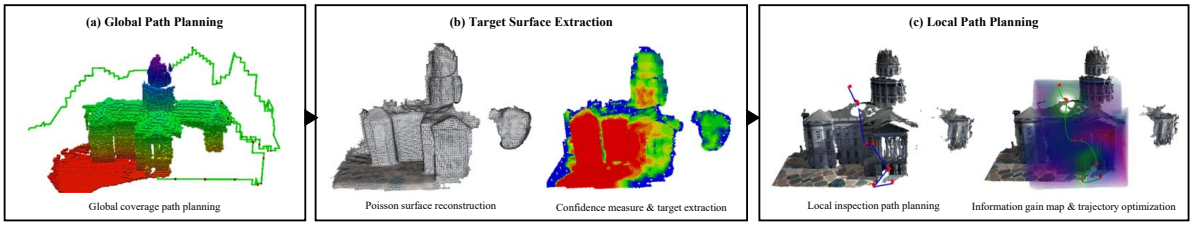

Fig. 4. Overview of proposed path-planning method. The proposed method is composed of three steps. (a) First, a global path is determined by computing the global coverage of unexplored regions. (b) Second, target surfaces (gray arrow points) are extracted by evaluating the reconstruction quality. (c) Third, local path planning provides an inspection path for the target surfaces. The inspection path is optimized to improve the performance of MVS.

according to the updated pose graph. We employ the surfel deformation method of [39] for the surface model. Instead of using a deformation graph [40], the method individually transforms the position of each surfel to preserve the global consistency with the SLAM module. After surfel deformation, we re-initialize the volumetric map to an unknown state and determine the occupied volumes directly from the deformed surfel position. We then update the free space by casting rays from each updated pose to the occupied volumes. The volumes that are already assigned to occupied state are excluded when free space is updated. The ray-casting is performed at twice the coarseness of the resolution of $\mathcal{M}$ for the fast update.

V. Path Planning Method

The aim of this study is to reconstruct a high-confidence surface model by exploring an unknown and bounded space as fast as possible. Algorithm 1 presents the pseudocode of the proposed path-planning method, which consists of three steps, as depicted in Fig. 4: *Global path planning*, *Target surface extraction*, and *Local path planning*. At every iteration, the proposed method first determines a global path that maximizes the coverage of the unexplored region (Section V.A). Then, it plans a local path that completes to reconstruct the surface model while maximizing the performance of the online MVS. To compute a local path, the proposed method extracts low-confidence surfaces (Section V.B) and determines a set of view configurations to acquire reference images of each low-confidence surface (Section V.C). Finally, it computes an optimal path that maximizes the performance of MVS when traversing all the reference viewpoints (Section V.D).

A. Global Path Planning

Our method divides the entire frontier in $\mathcal{M}$ into a set of clusters V_{front} (line 1) and computes the global coverage of the clusters. It explores the unknown region following the coverage sequentially. We formulate the problem of global coverage path planning as a *generalized traveling salesman problem* (GTSP) [41]. Let $q \in Q$ be a feasible view configuration in configuration space Q, and $\xi : [0,1] \rightarrow Q$ be a path. For each frontier cluster $V_i \in V_{front}$, we generate a set of view configuration samples Q_i in which more than a certain percentage of frontiers in V_i are visible (line 2). Given a set of sample sets $\{Q_1, ..., Q_N\}$, the GTSP algorithm

Algorithm 1 Proposed path planning algorithm

Input: Volumetric map $\mathcal{M}$, Surface model $\mathcal{F}$, and Current configuration q_{curr}.

/ Global path planning */*

1: $V_{front} \leftarrow FrontierClustering(\mathcal{M})$
2: $\{Q_1, ..., Q_N\} \leftarrow GlobalSampling(V_{front})$
3: $\{q_1, ..., q_N\} \leftarrow SolveGTSP(\{Q_1, ..., Q_N\}, q_{curr})$
4: $\{q_{NBV}, \xi_{global}\} \leftarrow GetPath(q_{curr}, q_1)$

/ Local path planning */*

5: **while** $q_{curr} \neq q_{NBV}$ **do**
6: **if** $TravelTime > \theta_{time}$ **then**
7: $\bar{X}_{target} \leftarrow GetTargetSurfPoints(\mathcal{F})$
8: $\{\dot{Q}_1, ..., \dot{Q}_N\} \leftarrow LocalSampling(\bar{X}_{target}, \xi_{global})$
9: $\{\dot{q}_1, ..., \dot{q}_N\} \leftarrow SolveGTSP(\{\dot{Q}_1, ..., \dot{Q}_N\}, q_{curr}, q_{NBV})$
10: $\{Q_{ref}, \xi_{local}\} \leftarrow GetPath(q_{curr}, \{\dot{q}_1, ..., \dot{q}_N\}, q_{NBV})$
11: $\xi^*_{local} \leftarrow OptmizePath(\bar{X}_{target}, Q_{ref}, \xi_{local})$
12: **end if**
13: $MoveToward(\xi^*_{local})$
14: $Update(\mathcal{M}, \mathcal{F}, q_{curr})$
15: **end while**

selects a representative point q_i from each sample set Q_i and obtains the shortest tour, departing from a current view q_{curr} and visiting all the representative points q_i (line 3). Paths are generated using the A* planner, which examines the minimum Euclidean distance among visiting points. Given the resulting coverage path, we select the first sample q_1 to be the next view configuration q_{NBV} and compute the global path ξ_{global} to q_{NBV} (line 4).

B. Target Surface Extraction

Our method first reconstructs a tentative surface model $\bar{\mathcal{F}}$ from the surfels in $\mathcal{F}$ using the screened Poisson reconstruction algorithm [42]. We represent the reconstructed surfaces in $\bar{\mathcal{F}}$ as a set of surface points X, where each point $x \in X$ contains 3D position and normal values. The method then groups adjacent surface points by Euclidean clustering. Let $X_j \subset X$ be a set of clustered surface points and $\bar{x}_j$ be the averaged surface point of X_j. We define the confidence of $\bar{x}_j$ as the average weight $\bar{w}_j$ of neighboring surfels in $\mathcal{F}$. Finally, the averaged surface points whose confidence values are lower than $\theta_{thr-conf}$ are determined as the target surface points $\bar{X}_{target}$ (line 7).

C. Local Inspection Path Planning

This section describes a method to compute an inspection path that provides coverage of the target surfaces. For each

target surface point $\bar{x}_j$, the method generates a set of view configurations $\dot{Q}_j$ that observe the target point $\bar{x}_j$ by inversely composing a view frustum from $\bar{x}_j$ to its normal direction [23] (line 8). We reject a sample configuration $\dot{q}_j$ if the distance of the path via $\dot{q}_j$ is γ times (1.3 in this paper) larger than the distance of the direct path from q_{curr} to q_{NBV}. Similar to Section V.A, the GTSP algorithm is used to determine a tour starting from q_{curr}, visits exactly one sample $\dot{q}_j$ per sample set $\dot{Q}_j$, and ends at q_{NBV} (line 9). The local path ξ_{local} is computed by sequentially connecting the selected samples (line 10). We refer to the selected tour set $\{\dot{q}_1, ..., \dot{q}_N\}$ as the reference configuration set Q_{ref}, which is used to reference views for each target point.

D. Trajectory Optimization

After determining a local inspection path for the target surfaces, our method refines the path to maximize the MVS performance (line 11). The following subsections introduce stereo-pair heuristics for predicting the reconstruction quality of MVS and describe how to apply these to the trajectory optimization problem.

1) Multi-view Stereo Heuristics: Given a stereo-pair of a reference view configuration q_{ref} and a source view configuration q_{src}, the reconstruction quality of a target surface $\bar{x}$ is related to geometric factors, such as *parallax*, *relative distance*, and *focus*.

Parallax: There is a trade-off between the triangulation accuracy and the matchability according to the parallax of a stereo pair. Let α be a parallax angle between pair sets in a stereo pair. We describe the informativeness of α to reconstruct the correct surface as a score function [10]:

$$f_{prx}(\alpha) = \exp\left(-\frac{(\alpha - \alpha_0)^2}{2\sigma_{prx}^2}\right) \tag{2}$$

where α_0 is the desired parallax angle, which is heuristically determined as $15°$, and σ_{prx} is a constant value.

Relative distance: The image patches from the reference and source images must have a similar resolution for accurate stereo matching [2]. We assume that the views at the same distance to a surface have the same resolution of the surface. A score function regarding the relative distance is defined as

$$f_{rd}(dist_{src}, dist_{ref}) = \frac{\min(dist_{src}, dist_{ref})}{\max(dist_{src}, dist_{ref})} \tag{3}$$

where $dist_{src}$ and $dist_{ref}$ denote the distance between $\bar{x}$ and q_{src}, and the distance between $\bar{x}$ and q_{ref}, respectively.

Focus: The surface region preferably projects to the principal point of the source image to reduce the reprojection error in triangulation [29] [30]. Let $r_{\overline{co}}$ and $r_{\overline{cx}}$ be rays from a camera center c of a source image to the principal point o and to a surface point $\bar{x}$, respectively. We define a penalizing function for large angle β between the rays $r_{\overline{co}}$ and $r_{\overline{cx}}$ as

$$f_{foc}(\beta) = \exp\left(-\frac{\beta^2}{2\sigma_{foc}^2}\right) \tag{4}$$

where σ_{foc} is a constant value.

Integration: We integrate the heuristics into a score function that predicts the reconstruction quality of MVS:

$$f_{src}(q_{src}, q_{ref}, \bar{x}) = f_{vis} \cdot f_{prx} \cdot f_{rd} \cdot f_{foc} \tag{5}$$

where f_{vis} is the visibility function that returns the value 1 if $\bar{x}$ is visible from q_{src}, and the value 0 otherwise.

2) Informative Path Planning: A set of disjoint path segments $\boldsymbol{\xi} = \{\xi_1, ..., \xi_N\}$ is based on each reference configuration in Q_{ref}. Each segment ξ_s is a path connecting the consecutive reference view configurations. Let $IG(\xi_s)$ be a function that represents the information gathered along ξ_s and $TIME(\xi_s)$ be the corresponding travel time. The optimal path is determined by solving the following problem:

$$\begin{aligned} \boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmax}} \sum_{\xi_s \in \boldsymbol{\xi}} \frac{IG(\xi_s)}{TIME(\xi_s)}, \\ \text{s.t } TIME(\xi_s) \leq B_s \text{ for every segment } s \end{aligned} \tag{6}$$

where B_s is a time budget of segment s. We define a budget as $B_s = \gamma' \times TIME(\bar{\xi}_s)$, where γ' is a constant value, and $\bar{\xi}_s$ is the shortest path from the starting point to the end point of ξ_s. An information gain function is defined as

$$IG(\xi_s) = \sum_{q_i \in \xi_s} \sum_{q_r \in Q_{ref}} f_{src}(q_i, q_r, \bar{x}_r) \tag{7}$$

where q_i is a discrete configuration along ξ_s, q_r is a reference view configuration, and $\bar{x}_r$ is the target surface of q_r. Eq. 6 is an *informative path-planning problem* that can be solved as an optimization problem. To solve the problem, we employ the local optimization step in [43], which uses the covariance matrix adaptation evolution strategy [44]. The strategy is based on a Monte Carlo method; it uses the process of repeated random sampling to estimate the solution.

E. Reference and Source Image Selection

Our method consistently determines reference images every time a keyframe is extracted. The keyframes are consistently extracted at a regular frame interval. Given a reference image, N keyframes (10 in this paper) are selected for a source image set by evaluating the score function Eq. 5 for each path segment ξ_s. The q_{NBV} at the end of ξ^*_{local} does not have a target point; thus, it uses N-neighbor keyframes that share most map-point observations for source images.

VI. EXPERIMENTAL RESULTS

In this section, we conducted the simulation experiments to evaluate the performance of the proposed method. A firefly hexacopter MAV was used as a mobile robot in the RotorS simulation environment [45]. The forward-looking stereo camera, mounted on the MAV, had a pitch angle $5°$ downward and a field of view $[60°, 90°]$. It captured images with a resolution 752×480px. We used a stereo version of the ORB-SLAM [46] to obtain a metric scale of the motion and map points. For a reliable pose estimation, we restricted the maximum translational speed $0.5m/s$ and rotational speed $0.25m/s$ and covered an textured scene to the ground. We used only the left images on the stereo camera for MVS computation.

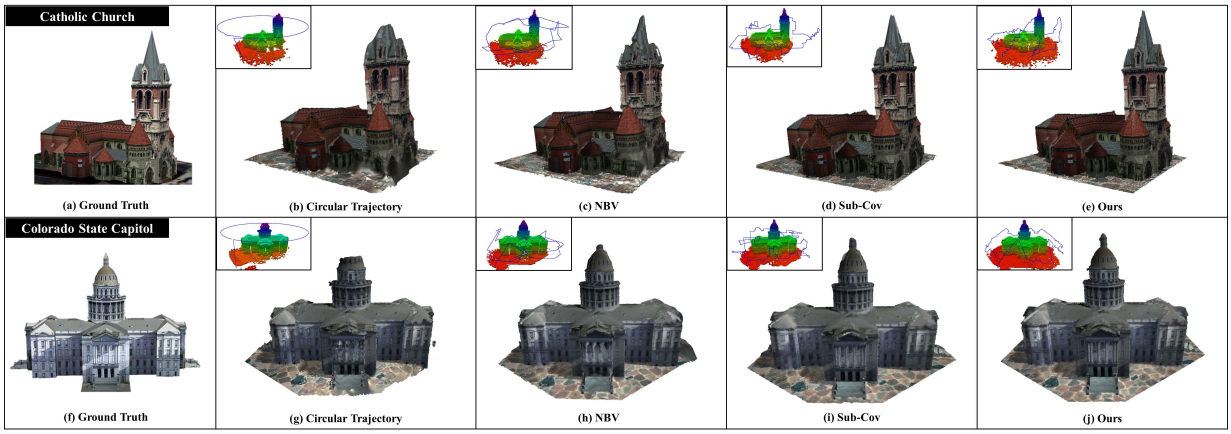

Fig. 5. Experimental results of scenario 1 (upper) and scenario 2 (lower). Reconstructed 3D models and volumetric maps with trajectories taken by the MAV at the end of executions of (b, g) circular trajectory, (c, h) NBV [5], (d, i) Sub-Cov [9], and (e, j) our method.

TABLE I
EXPERIMENTAL RESULTS OF TWO SCENARIOS.

	Methods	Time (*min*)	Path Length (m)	Error (m)	Comp. $0.05m$ (%)	Comp. $0.10m$ (%)	Comp. $0.15m$ (%)
Scenario 1	Circular	12.68	239.6	0.1596	29.33	52.77	66.64
	NBV	29.60	643.8	0.0987	53.06	71.97	82.19
	Sub-Cov	28.72	654.5	0.0805	**57.45**	**73.98**	82.71
	Ours	**18.37**	**383.1**	**0.0746**	56.85	72.35	**84.46**
Scenario 2	Circular	15.60	290.3	0.1560	30.10	46.20	59.70
	NBV	43.12	891.6	0.1273	48.38	65.77	77.67
	Sub-Cov	46.65	1016.5	0.0961	52.24	69.60	82.92
	Ours	**29.64**	**498.2**	**0.0736**	**62.51**	**80.07**	**86.75**

There are two-simulation scenarios [47]: a structure with highly textured surfaces (*Catholic Church:* $36 \times 28 \times 30m^3$) and a structure with relatively less-textured surfaces (*Colorado State Capitol:* $50 \times 46 \times 30m^3$). The proposed method was compared with the **NBV** [5] and submodular coverage (**Sub-Cov**) [9] methods. These methods use the explore-then-exploit strategy; after constructing an initial coarse model from a default trajectory, they compute the coverage path [9] or NBVs [5] for the coarse model. The NBV method iteratively determines the best viewpoint online from a partial reconstruction. The Sub-Cov method computes the coverage path of the initial model offline. Both the initial and final models were constructed using our online modeling system. We performed an initial scan of a **circular trajectory** around the target space with a camera pitch of $20°$. We tuned the travel budget of Sub-Cov to obtain the best modeling performance in each scenario. Every reconstructed surface model was post-processed using the Poisson surface reconstruction.

The performance of the proposed method, compared to the other methods, was evaluated from two different perspectives: path efficiency and modeling quality. To evaluate the path efficiency, we computed the completion time and total path length. The modeling quality refers to the accuracy and completeness of the surface model [48]. The accuracy was estimated by calculating the mean errors. The completeness was defined as the percentage of surface points that have distance error smaller than thresholds. Fig. 5 depicts the paths and the corresponding models of the best trial. Table I presents the average results of five executions. The results of NBV and Sub-Cov show the cumulative time and path length during the whole explore-then-exploit process.

Our method had the best performances in terms of the path efficiency. Our method provided an efficient coverage path that did not frequently overlap while taking less travel time without an initial scanning. Moreover, even with these short trajectories, our method achieved the best modeling performances in terms of the accuracy and completeness. Particularly in Scenario 2, our method outperforms the others in terms of overall performance. The modeling system does not guarantee a complete and accurate depth estimate of a reference image; therefore, several factors such as reconstruction quality and MVS heuristics should be considered online. However, NBV and Sub-Cov do not consider these factors during path planning. Especially, NBV focuses only on the viewpoint that covers the largest surface area while disregarding minor surfaces, so their reconstructed models can be incomplete. Our method, on the other hand, focuses on completing the insufficiently reconstructed regions by examining the completeness of reconstructed surfaces. It also optimizes a path to obtain the best reference and source images; these approaches enhance the modeling performances.

VII. CONCLUSION

We proposed an online MVS algorithm for view path planning of a large-scale structure. The proposed method computes a view path using the online feedback based on the quality of the reconstructed surface. The computed path provides full coverage of low-quality surfaces while maximizing the reconstruction performance of MVS. To the best of our knowledge, this is the first work that implements an exploration planning approach to model unknown structures using an online MVS system.

REFERENCES

[1] B. Hepp, M. Nießner, and O. Hilliges, "Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Transactions on Graphics*, vol. 38, no. 1, p. 4, 2018.

[2] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 501–518.

[3] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[4] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3260–3269.

[5] R. Huang, D. Zou, R. Vaughan, and P. Tan, "Active image-based modeling with a toy drone," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[6] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Autonomous Robots*, vol. 40, no. 6, pp. 1059–1078, 2016.

[7] W. Jing, J. Polden, W. Lin, and K. Shimada, "Sampling-based view planning for 3d visual coverage task with unmanned aerial vehicle," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1808–1815.

[8] M. D. Kaba, M. G. Uzunbas, and S. N. Lim, "A reinforcement learning approach to the view planning problem," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5094–5102.

[9] 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 *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5324–5333.

[10] N. Smith, N. Moehrle, M. Goesele, and W. Heidrich, "Aerial path planning for urban scene reconstruction: a continuous optimization method and benchmark," in *SIGGRAPH Asia 2018 Technical Papers*. ACM, 2018, p. 183.

[11] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2609–2616.

[12] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2320–2327.

[13] K. Wang, W. Ding, and S. Shen, "Quadtree-accelerated real-time monocular dense mapping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[14] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[15] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 333–356, 1988.

[16] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.

[17] N. J. Sanket, C. D. Singh, K. Ganguly, C. Fermüller, and Y. Aloimonos, "Gapflyt: Active vision based minimalist structure-less gap detection for quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2799–2806, 2018.

[18] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, vol. 11, no. 2011, 2011, pp. 127–136.

[19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[20] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2135–2142.

[21] Z. Meng, H. Qin, Z. Chen, X. Chen, H. Sun, F. Lin, and M. H. Ang Jr, "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.

[22] S. Song and S. Jo, "Online inspection path planning for autonomous 3d modeling using a micro-aerial vehicle," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6217–6224.

[23] ——, "Surface-based exploration for autonomous 3d modeling," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[24] C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, and G. Reitmayr, "Online feedback for structure-from-motion image acquisition." in *British Machine Vision Conference (BMVC)*, vol. 2, 2012, p. 6.

[25] S. Haner and A. Heyden, "Covariance propagation and next best view planning for 3d reconstruction," in *European Conference on Computer Vision (ECCV)*. Springer, 2012, pp. 545–556.

[26] "Pix4d. pix4dcapture." 2017. [Online]. Available: http://pix4d.com/product/pix4dcapture

[27] A. P. U. Manual, "Professional edition," *Aplastic Anemia (Hypoplastic Anemia)*, 2014.

[28] C. Hoppe, A. Wendel, S. Zollmann, K. Pirker, A. Irschara, H. Bischof, and S. Kluckner, "Photogrammetric camera network design for micro aerial vehicles," in *Computer Vision Winter Workshop (CVWW)*, vol. 8, 2012, pp. 1–3.

[29] O. Mendez Maldonado, S. Hadfield, N. Pugeault, and R. Bowden, "Next-best stereo: extending next best view optimisation for collaborative sensors," *British Machine Vision Conference (BMVC)*, 2016.

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

[31] C. Forster, M. Pizzoli, and D. Scaramuzza, "Appearance-based active, monocular, dense reconstruction for micro aerial vehicle," in *2014 Robotics: Science and Systems Conference*, no. EPFL-CONF-203672, 2014.

[32] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *2013 IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1385–1392.

[33] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[34] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *2013 IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1449–1456.

[35] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do, "Fast global image smoothing based on weighted least squares," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5638–5653, 2014.

[36] D. Gallup, J.-M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1418–1425.

[37] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation," *Computational Geometry*, vol. 22, no. 1-3, pp. 21–74, 2002.

[38] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo-stereo matching with slanted support windows." in *British Machine Vision Conference (BMVC)*, vol. 11, 2011, pp. 1–11.

[39] K. Wang, F. Gao, and S. Shen, "Real-time scalable dense surfel mapping," in *2019 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6919–6925.

[40] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 80, 2007.

[41] B. Hu and G. R. Raidl, "Effective neighborhood structures for the generalized traveling salesman problem," in *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer, 2008, pp. 36–47.

[42] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics*, vol. 32, no. 3, p. 29, 2013.
[43] M. Popović, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran, "Online informative path planning for active classification using uavs," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5753–5758.
[44] N. Hansen, "The cma evolution strategy: a comparing review," in *Towards a new evolutionary computation*. Springer, 2006, pp. 75–102.
[45] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors—a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.
[46] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
[47] "Google's 3d warehouse." [Online]. Available: http://3dwarehouse.sketchup.com/
[48] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 78, 2017.