View Path Planning via Online Multiview Stereo for 3-D Modeling of Large-Scale Structures

Soohwan Song, Daekyum Kim, and Sunghee Choi

***Abstract*—This study addresses a view-path-planning problem during 3-D scanning of a large-scale structure based on multiview stereo (MVS) for unmanned aerial platforms. Recently, most studies have adopted an explore-then-exploit strategy for 3-D scanning. The strategy first generates a coarse model from a simple overhead scanning and then plans an inspection path to cover the entire surface of the coarse model. However, even though the inspection path may be optimal, it is difficult to guarantee a complete and accurate reconstruction result due to defective factors of MVS such as occlusions, textureless surfaces, and insufficient parallaxes. Furthermore, the entire procedure of this strategy is inefficiently slow because of path redundancies and long MVS processing time. Therefore, we propose a novel view-path-planning method for 3-D scanning based on an online MVS reconstruction algorithm. The suggested method incrementally reconstructs the target model online and iteratively plans view paths by analyzing the current partial reconstructions. The method continuously analyzes the quality of the model and detects inaccurately reconstructed surfaces. It then plans an inspection path that provides a complete coverage of the detected surfaces while maximizing the performance of MVS. This method can construct a complete 3-D model in a single scanning trial without the need for rescanning. Extensive experiments show that our method outperforms the other state-of-the-art methods, especially in terms of the model completeness of complex structures.**

***Index Terms*—3-D reconstruction, active sensing, multiview stereo, unmanned aerial vehicles, view path planning.**

I. Introduction

THE demand for high-quality 3-D models is continuously increasing in many industrial applications, such as geospatial applications, structural inspection, and content creation in computer games. To obtain high-quality 3-D models, *multiview stereo* (MVS) methods [1]–[3] have been widely used. MVS is an offline approach that reconstructs a 3-D model by batch processing a collection of calibrated images captured from different viewpoints. MVS methods are particularly effective for large-scale 3-D modeling because they can estimate a wide range of depths and reconstruct detailed and dense 3-D shapes of large-scale structures.

To acquire images for MVS reconstruction, microaerial vehicles (MAVs) equipped with cameras could be a suitable platform due to their high maneuverability and ability to reach almost any vantage point. However, there are tradeoffs to consider for MVS reconstruction through MAVs. MAVs have limited battery capacity; it requires to collect images in a limited time. On the other hand, the completeness and quality of MVS reconstructions heavily depend on the set of images that densely represent the target structure [2], [4], [5]. Therefore, an efficient view-path-planning algorithm is required to successfully reconstruct large-scale structures. Recently, several commercial applications for flight planning [6], [7] have been developed to facilitate 3-D scanning tasks. These applications produce simple circular or lawnmower trajectories within a safe overhead area for scanning. However, these approaches do not consider the geometry of target structures; therefore, the vehicles cannot scan occluded areas and thus produce incomplete 3-D models. To address this issue, the *explore-then-exploit* method [8]–[11] has been proposed. This method first constructs an initial coarse model by scanning an entire scene using a simple fixed trajectory within a safe area. Then, it computes an inspection path to cover the entire surface of the coarse model. Based on the computed inspection path, the method performs a rescanning and reconstructs the final 3-D model of the target structure.

Although the explore-then-exploit methods can provide an optimal inspection path for a given coarse geometry, their modeling performance can be degraded due to several reasons. First, these methods sometimes produce overlapping trajectories, which is inefficient in time because the same areas are repeatedly scanned. Second, even if the acquired images provide full coverage of the target structure, MVS methods do not guarantee the reconstruction of accurate and complete 3-D models when textureless scenes, short baseline distances, or occlusions are present. Finally, MVS methods generally take a long time to process input images, which makes the entire procedure of these methods very slow.

We present a novel view-path-planning method for 3-D scanning based on online MVS reconstruction. The proposed

Manuscript received September 16, 2020; revised January 18, 2021; accepted April 21, 2021. Date of publication June 10, 2021; date of current version February 8, 2022. This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2016R1D1A1B01013573, and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) funded by the Korea Government under Grant 2019-0-01158. This paper was recommended for publication by Associate Editor S. Shen, and Editor F. Chaumette upon evaluation of the reviewers' comments. *(Corresponding author: Sunghee Choi.)*

Soohwan Song is with the Intelligent Robotics Research Division, Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea (e-mail: soohwansong@etri.re.kr).

Daekyum Kim and Sunghee Choi are with the School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: daekyum@kaist.ac.kr; sunghee@kaist.ac.kr).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2021.3083197.

Digital Object Identifier 10.1109/TRO.2021.3083197

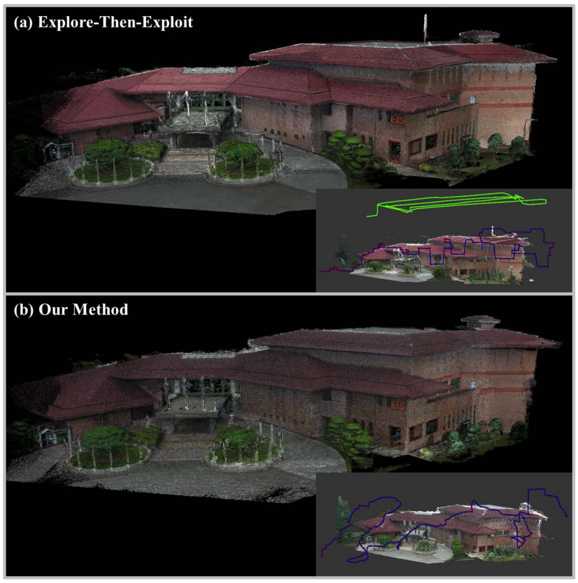


Fig. 1. Reconstructed 3-D models of the real-world scene (Auditorium) with trajectories taken by the MAV. (a) Reconstruction result of the explore-then-exploit method [8] and offline MVS [3]. It generates a coarse model from a default trajectory (in green) and then plans the inspection path (in blue). It takes 10.5 h in total (3.3 h for the coarse model and 7.2 h for detailed reconstruction) to process all the acquired images. (b) Reconstruction result of our view-path-planning method with online MVS. It incrementally constructs the 3-D model in real time and completes the entire modeling process in a single scanning trial.

view-planning method continuously detects incompletely reconstructed surfaces by analyzing the quality of the 3-D model and plans view paths to scan them. The method computes the best viewpoints to acquire reference and source images using MVS heuristics. Based on the computed viewpoints, it provides an optimal view path that satisfies the dual requirements of scanning the incomplete surfaces and maximizing the reconstruction performance of MVS. To enable the iterative view path planning, our method incrementally updates 3-D models in real time from an online MVS system, unlike existing offline methods [8]–[11]. The proposed online MVS system estimates camera poses using the simultaneous localization and mapping (SLAM) module [12] in real time and computes depth maps of keyframes using the deep-learning-based MVS method [13]. The system deals with excessive outliers in the estimated depths by applying several filtering steps, including photometric and geometric consistency checks. It continuously integrates estimated depths into a large-scale 3-D model using surfel-based mapping [14]. Finally, the proposed 3-D modeling framework including online MVS reconstruction and view-path planning can generate an accurate and complete 3-D reconstruction of large structures in a single exploration trial without the need for rescanning, unlike the explore-then-exploit methods (see Fig. 1).

A. Contributions and Outline

The major contributions of this study can be summarized as follows:

1) Unlike existing explore-then-exploit approaches, we propose a comprehensive framework for autonomous 3-D modeling, which features online MVS reconstruction, repetitive reconstruction quality evaluation, and exploration planning.
2) We propose a view-path-planning method that efficiently explores the unknown areas and simultaneously analyzes the quality of reconstructed surfaces. The computed path provides full coverage of low-quality surfaces while maximizing the performance of MVS reconstruction.
3) We present an online 3-D-modeling system based on SLAM and deep-learning-based MVS [13]. This system applies strict outlier filtering and constructs a globally consistent model using surfel-based mapping [14].
4) The proposed method is evaluated in both simulated and real-world environments. The effectiveness and applicability of the proposed method were evaluated and compared with those of state-of-the-art methods.

A preliminary version of this article has been presented in [15]. In this study, we used the deep-learning-based MVS method [13] instead of monocular dense mapping [16] for depth estimation. The deep-learning-based method can estimate a high-quality depth map in real time and has better reconstruction performance than the monocular mapping method. We provide a more detailed explanation of the proposed framework and present a thorough evaluation of it using various experimental scenarios, including two real-world environments. Furthermore, we address the performance of the proposed online MVS experimentally by comparing it with the existing offline MVS method [3].

The remainder of this article is structured as follows. Section II presents related work on many types of view-path-planning methods. Section III describes the target problem and an overview of the proposed framework. The proposed online 3-D modeling system is detailed in Section IV, and the view-path-planning method is detailed in Section V. Section VI describes simulation and real-world experiments. Finally, we discuss our findings and the limitations of our study in Section VII and summarize the contributions of this study in Section VIII.

II. Related Works

The problem of determining the optimal viewpoints for reconstructing 3-D models of scenes or objects is known as the active vision or view-path-planning problem. This problem has been extensively addressed for more than two decades [17]–[19]. Different types of 3-D modeling methods have been developed and used in various view-path-planning studies. Table I presents a summary of previous view-path-planning studies based on the types of 3-D modeling methods, namely, *volumetric mapping*, *dense surface mapping*, and MVS. The following sections present literature reviews on each type of modeling method.

A. Volumetric Mapping

The volumetric mapping method [20], [21] encodes the state of an environment into an octree-based occupancy grid volumetric map. The volumetric map explicitly represents not only the occupied volumes but also the free and unknown volumes. It

TABLE I
PREVIOUS VIEW-PATH-PLANNING STUDIES ACCORDING TO THEIR 3-D MODELING METHODS

Modeling method	**Approach**	**References**	**Online/ offline**	**Dense/sparse modeling**	**Modeling target**	**Descriptions**
Volumetric mapping [20] [21]	NBV planning	[22] [23] [24]	Online	Sparse	Indoor/outdoor scene	Frontier-based NBV estimation
		[25] [26]	Online	Sparse	Indoor/outdoor scene	Use information-theoretic measures for NBV evaluation
		[27] [28]	Online	Sparse	Indoor / outdoor scene	Machine-learning-based NBV estimation
	Next-best trajectory	[29] [30] [31]	Online	Sparse	Indoor/outdoor scene	Evaluate view sequences or optimize view paths toward an NBV
	Inspection planning	[32] [33]	Online	Sparse	Indoor/outdoor scene	Online inspection method for complete 3D modeling
Dense surface mapping [34] [14]	Volumetric method	[35] [36] [37]	Online	Dense	Small-scale object	Analyze a volumetric model for NBV evaluation
	Surface-based method	[38] [39] [40] [41]	Online	Dense	Small-scale object	Consider the shape and quality of reconstructed surfaces
	Integrated method	[42]	Online	Dense	Small-scale object	Integration of surface-based and volumetric methods for small-scale object modeling
		[43]	Online	Dense	Outdoor scene	Surface-based exploration for outdoor scene modeling
Multi-view stereo [1] [2] [3]	Simple fixed trajectory	[6] [7]	Offline	Dense	Outdoor scene	Simple circular or zigzag trajectories in a safe overhead area
	Explore-then-exploit	[8] [9] [10] [11] [44]	Offline	Dense	Outdoor scene	Plan an inspection path for initially estimated coarse model
	NBV planning based on SfM	[45] [46]	Offline	Sparse	Outdoor scene	NBV determination based on sparse SfM feature points
	View planning based on online MVS	[47] [48]	Online	Dense	Indoor scene	Determine the best stereo pair to maximize stereo matching performance
		[49]	Online	Dense	Single depth map	Informative motion trajectory for single depth estimation
		Proposed method	**Online**	**Dense**	**Outdoor scene**	**Exploration path planning via online MVS system**

facilitates the visibility check for a specific volume and provides direct access to the free space in the environment. These functions are essential for the collision-free navigation of mobile robots. Therefore, the volumetric mapping method is mainly used for the exploration task in environment modeling.

Most exploration methods have focused on the *next best view* (NBV) problem, in which the best viewpoint for incrementally completing a model is determined using feedback from the current partial reconstruction. NBVs are determined using various metrics, such as the number of frontiers [22]–[24], and information-theoretic [25], [26] or machine-learning-based measures [27], [28]. Recently, some studies [29]–[31] have attempted to compute the most informative view paths rather than a single optimal viewpoint. They evaluated a set of candidate view paths to determine the most informative path. Charrow *et al.* [29] proposed a method that determines the most promising path from global paths and local motion primitives and optimizes the path according to an information-theoretic objective. Wang *et al.* [31] planned an exploration path by building a 3-D topological road map and used a potential field-based local planner to collect more information along the exploration path. Several methods [32], [33] used an inspection strategy to model an unknown environment. Unlike conventional inspection problems [50], [51], which assume that the model is known in advance, they addressed online inspection planning, starting from partially known models. These methods completely scan local frontiers and thus improve the completeness of the constructed volumetric models.

All these existing methods were based on depth sensors. However, the sensing ranges of most depth sensors are relatively shorter than 15 m. Therefore, robots need to get close to the unvisited viewpoints, leading to inefficiencies in exploration trajectories. On the other hand, our method can reconstruct long-range scenes about 50 m or more using the online MVS, thus enabling to plan exploration paths more efficiently.

B. *Dense Surface Mapping*

Dense-surface-mapping methods [14], [34] construct dense models by directly accumulating RGB-D data obtained from a depth sensor. Most of the mapping methods use a point cloud registration method such as the iterative closest point algorithm for data accumulation. NBV approaches have also been frequently used for the view-path-planning problem in dense surface mapping. According to Scott *et al.* [18], NBV approaches can be classified into two categories: *volumetric* and *surface-based*.

Volumetric methods [35]–[37] analyze the spatial information from a volumetric model to determine the NBV for dense surface mapping. Vasquez-Gomez *et al.* [35] analyzed a set of frontier voxels and determined the NBV for dense 3-D modeling of small-scale objects. Delmerico *et al.* [36] provided several metrics to quantify the volumetric information contained in the voxels. Even if the volumetric model is complete, it does not necessarily mean that the quality of the reconstructed surface is also perfect. It is challenging to represent the reconstructed surfaces of very complex structures as only free or occupied volumes in volumetric models. Therefore, volumetric methods are sometimes not appropriate for completing precise dense 3-D models and may produce poorly reconstructed surfaces.

Surface-based methods [38], [40], [41] determine the NBV by analyzing the shape and quality of the reconstructed surfaces. These methods concentrate on completing the surface model, which is represented as a set of mesh surfaces or a high-resolution point cloud. Chen and Li [38] proposed a mathematical model for predicting the surface trend of an object to determine the NBV for the reconstruction of simple and smooth surfaces. Wu *et al.* [40] estimated the confidence map, which represents the completeness and smoothness of the constructed Poisson isosurfaces. A confidence map is used to guide the computation of an NBV. Lee *et al.* [41] proposed a 3-D scanning method for mechanical parts. The method determined an NBV by detecting surface primitives and the underlying shape of the ground-truth model from the scanned partial data.

Some studies [42], [43] proposed a combination of surface based and volumetric methods. Kriegel *et al.* [42] presented a method to compute the best-scan-path for small-scale object modeling. They generated scan path candidates by estimating the surface trends and computed the information gain of each scan path from the volumetric model. Song and Jo [43] proposed an integrated approach of surface-based and volumetric methods for modeling a large-scale structure. This approach used a volumetric map for fast exploration of an unknown area and analyzed the surface model to improve the quality of the reconstructed surfaces simultaneously. Similar to this approach [43], our method also plans a path by analyzing both a volumetric map and a surface model, but also accounts for the heuristic MVS information to improve the performance of the MVS.

C. Multi-View-Stereo

MVS algorithms [1]–[3] reconstruct a dense 3-D model by computing the stereo correspondences of calibrated images in a batch. MVS algorithms have been frequently used for modeling large-scale structures because they can estimate a wide depth range of the target scene. Some methods [6], [7] acquired a set of images for MVS reconstruction using simple circular or zigzag trajectories in a safe overhead area. However, these methods do not provide complete coverage of the target scene. Therefore, as noted earlier, many studies that use MVS reconstruction have commonly adopted the explore-then-exploit method [8]–[11], [44]. Several methods [8], [9] formulated inspection path planning as a submodular optimization problem. Hepp *et al.* [9] used a volumetric model to evaluate the coverage trajectories. They defined the information gain of a trajectory as the number of visible voxels in the volumetric model and computed a coverage path using submodular optimization. Roberts *et al.* [8] defined the information gain of a scanning trajectory as the total covered region of hemispheres around each surface point. Their method finds an optimal coverage trajectory that scans entire surfaces from diverse viewing directions by solving an orienteering problem. Huang *et al.* [44] presented a relatively fast MVS algorithm for coarse model reconstruction instead of dense modeling. They computed the NBVs by evaluating the coverage of the coarse model.

Several view-planning methods [45], [46] have been proposed to determine the best viewpoints by analyzing a sparse point cloud using *structure-from-motion* (SfM) [52], [53]. The SfM aims to obtain camera poses and sparse point cloud using a set of collected images before performing MVS reconstruction. The view-planning methods [45], [46] compute view paths by using the sparse point cloud to acquire the images necessary for MVS. Hoppe *et al.* [45] presented an online SfM framework that provides a human operator with visual feedback of the modeling quality. The modeling quality is represented by the ground sampling distance and image redundancy. Mauro *et al.* [46] proposed a view importance measure for the NBV planning and image selection. The view importance indicates the significance of a viewpoint for 3-D reconstruction and is estimated through several quality features extracted from a sparse SfM point cloud.

There have been only a few approaches [47]–[49] that consider online MVS reconstruction. Mendez *et al.* [47], [48] reconstructed a dense scene in real time using deep-learning-based stereo matching [54]. They presented a next-best stereo method that selects a stereo pair by jointly optimizing the baseline and vergence to maximize the performance of pairwise stereo matching. Forster *et al.* [49] addressed the view planning problem by acquiring an informative motion trajectory for monocular dense depth estimation [16]. They formulated a depth measurement uncertainty that accounts for both the scene structure and texture to evaluate a motion trajectory. These methods [47]–[49] focused only on estimating local depth maps and did not consider global model reconstruction. In contrast, our proposed framework addresses not only the local path for local depth mapping but also the global path for constructing the entire model.

III. Problem Description and System Overview

The problem considered in this study is exploring an unknown and spatially bounded 3-D space $V \subset \mathbb{R}^3$ using a MAV while reconstructing high-quality 3-D models of structures in the space. The MAV is equipped with a forward-looking camera to acquire image frames. The acquired image frames are processed in real time to estimate the current localization and construct 3-D models. Our approach constructs two 3-D models: a volumetric map $\mathcal{M}$ and a surface model $\mathcal{F}$. The volumetric map $\mathcal{M}$ represents the workspace in three states (occupied $V_{occ} \subset V$, free $V_{free} \subset V$, and unknown $V_{ukn} \subset V$) and is constructed using the Octomap framework [20]. The free space information V_{free} can be directly accessed in the volumetric map for efficient

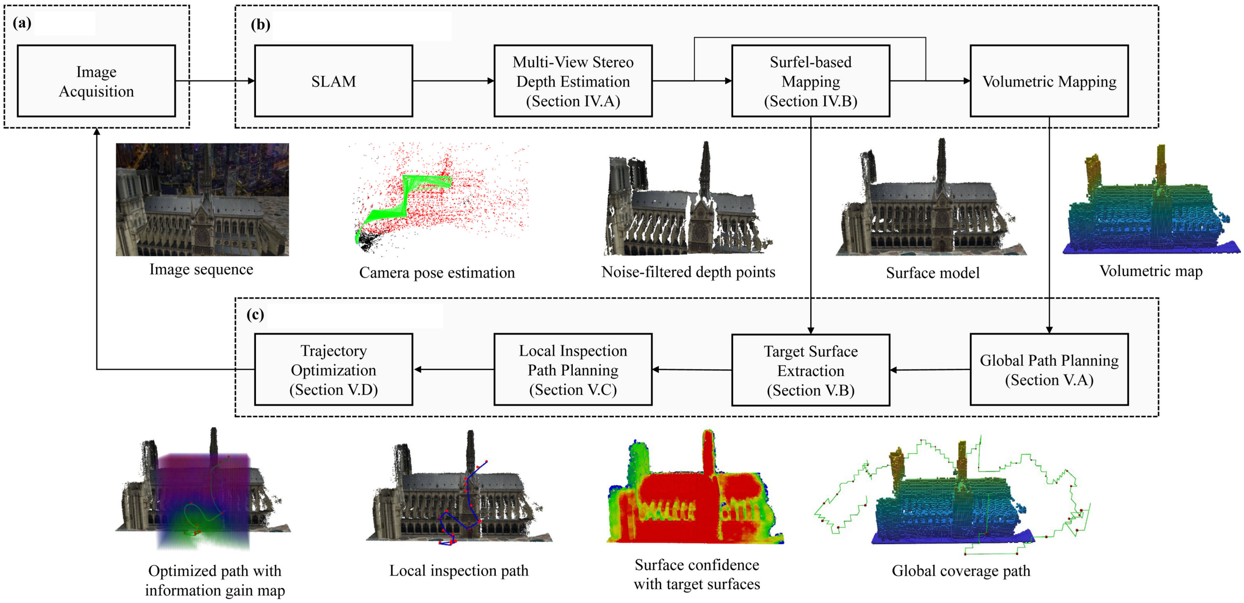


Fig. 2. Overview of the proposed 3-D modeling framework. (a) Image frames are captured from a camera mounted on the MAV. (b) The 3-D modeling module takes an image and estimates the camera pose of the frame using a SLAM system. It then infers the depth maps of selected frames and integrates them into a volumetric map and surface model. (c) The path-planning module analyzes the volumetric map and surface model to compute the global path and local inspection path, respectively. The local inspection path is then refined to maximize the performance of the MVS reconstruction. The MAV moves along the computed path while continuously scanning. These steps are repeated until the reconstructed model is completed.

planning of a collision-free path. The unknown space V_{ukn} represents an unexplored area. This is required for evaluating the utility of an exploration path. The surface model $\mathcal{F}$ explicitly represents densely reconstructed surfaces of the target structures. The surface model is constructed using a surfel-based mapping method [14]. This model comprises a collection of dense point primitives known as surfels that contain various surface information, such as the normal, confidence, and radius.

This work assumes that the MAV dynamics are differentially flat [30] and that the MAV configuration q is composed of an xyz-position and yaw orientation ψ with zero roll and pitch. Using the same assumptions as in [30], we limit the maximum translational speed v_{max} and the rotational speed $\dot{\psi}_{max}$ to small values for accurate state estimation and exact path following. Let Q be a feasible configuration space that is composed of all possible configurations. A path $\xi : [0, 1] \rightarrow Q$ is defined as a sequence of configurations. For collision-free navigation, the path should lie in free space V_{free}. The objective is to generate a path that satisfies both of the following objectives simultaneously: exploration of the entire unknown space in the volumetric map $\mathcal{M}$ within a short period, and reconstruction of a high-quality surface model $\mathcal{F}$ that densely covers the surfaces of target structures.

To solve this problem, we present a comprehensive 3-D modeling framework consisting of two functional modules: online 3-D modeling and path planning. Fig. 2 depicts the proposed framework. The 3-D modeling module builds a volumetric map and surface model of the environment using acquired image frames. This module first estimates the camera poses of the image frames using a SLAM system [12]. It then computes the depth maps of selected frames using an online MVS algorithm [13]. The computed depth maps are integrated into a volumetric map $\mathcal{M}$ and surface model $\mathcal{F}$ at the same time. The path-planning module computes the exploration paths to complete the 3-D models of the target structures. This planning module first analyzes the volumetric map and determines a global path with NBVs to explore a large unknown region. Then, the module plans a local inspection path that scans the low-confidence surfaces in the surface model. The planned local path is optimized to improve the performance of the MVS reconstruction. The MAV iteratively computes the optimized path and navigates along the path until the entire environment is completely explored.

IV. Online 3-D Modeling

Fig. 2(b) illustrates the proposed online 3-D modeling system. The system first estimates the camera poses of the obtained image sequences using the SLAM module and subsequently processes the online MVS to compute the local depth maps. The system uses a keyframe-based SLAM method [12] that computes the camera pose by estimating the sparse map points from selected keyframes. At regular frame intervals, image-pose pairs are consistently stored in a database and used as source images for stereo matching. When a new keyframe is extracted, the keyframe is set as the reference image for depth estimation.

A depth map of the reference image is estimated using the deep-learning-based MVS method [13]. This method efficiently estimates a high-quality depth map by applying a cascade cost-volume formulation. Unlike existing online depth-estimation algorithms [16], [55], which use subsequent sequential images as source images for stereo matching, our method considers every acquired frame in the image-pose pair database. Our method selects the best set of source images in an active manner by considering heuristic information, such as the baseline distance and triangulation angle, to improve the depth-estimation performance. (See Section V-E on active image selection).

The depth maps estimated online have relatively more outliers than those using the offline approaches [1], [3]. Therefore, our method applies several outlier-filtering steps. The processed depth maps are fused into volumetric map $\mathcal{M}$ and surface model $\mathcal{F}$ simultaneously. The surfel-based mapping method [14] is used for surface modeling. This method maintains the dense detailed geometric information about the large-scale point clouds. It can efficiently process graphics operations, such as rendering and surface deformation. Our system also processes loop closing by deforming the surface model and the volumetric map according to the updated pose graph in the SLAM module.

Next, we describe how our method estimates a depth map via deep-learning-based MVS (see Section IV-A), filters out the outliers in the depth map, constructs the surface model (see Section IV-B), and processes loop closing (see Section IV-C).

A. Depth Estimation

Given reference image I_{ref} and a set of source images $\mathbf{I}_{src} = \{I_1, \ldots, I_N\}$, we infer depth map D_{ref} of the reference image. We use the *cascade MVS network* (CasMVSNet) [13] for depth estimation. CasMVSNet extends the deep-learning-based MVS method, MVSNet [56], by using multiple small cost volumes instead of a single large cost volume to decrease the GPU memory consumption and computation time. CasMVSNet progressively regresses depth maps in a coarse-to-fine manner by reducing the depth range and number of hypothesis planes at each cost volume stage. This cascade approach makes it possible to obtain high-resolution depth maps in real time. Moreover, CasMVSNet provides one of the best reconstruction performances to date. Therefore, it is appropriate for our online modeling system.

CasMVSNet first extracts multiscale features from a feature pyramid network [57]. The extracted multiscale features are used to build cascade cost volumes at gradually finer scales using 3-D CNNs [56]. Each cost volume is regularized by a multiscale 3-D CNN and then converted into a probability volume by performing Softmax normalization along the depth direction. For each stage, CasMVSNet estimates a depth map by taking the expectation values on the probability volume and propagates the depth map into the following stages to initialize the hypothesis planes. Each stage refines the estimated depth map from the previous stage with a finer depth hypothesis and higher resolution. The final depth map is obtained from the output of the last stage.

Depth Postprocessing: CasMVSNet produces depth maps that include outliers in the background regions and occluded areas. Furthermore, the depth maps estimated online have relatively more outliers than those from offline estimations because of insufficient source views restricted to previously acquired frames. Therefore, it is necessary to remove outliers.

We consider *geometric* and *photometric consistencies* for depth map filtering. Geometric consistency measures the consistency of the predicted depth $\bar{D}$ between neighboring depth maps [1]. The depth maps $\{D_1, \ldots, D_N\}$ of the source images $\{I_1, \ldots, I_N\}$ are already known because the previous frames have been sequentially processed. We compute the discrepancy between the estimated depth $\bar{D}$ and $\{D_1, \ldots, D_N\}$ through reverse-projection [1]. We first convert $\bar{D}$ into a point cloud and reproject it to each source image I_{src}. We then calculate each relative depth difference of a pixel p between the projected depth $\bar{D}_{prj}(p)$ and original depth $D_{src}(p)$ in the source image I_{src} as

$$f_{rel-diff}(\bar{D}_{prj}(p), D_{src}(p)) = \frac{|\bar{D}_{prj}(p) - D_{src}(p)|}{D_{src}(p)}. \quad (1)$$

If the relative depth difference is lower than 0.01, we consider that the projected depth $\bar{D}_{prj}(p)$ and original depth $D_{src}(p)$ are consistent. Depths that do not satisfy three-view consistency are regarded geometrically inconsistent and filter them out.

Photometric consistency describes the quality of the stereo matching. Similar to the approach in [56], we estimate a confidence map to measure the photometric consistency. The confidence map can be estimated from the probability volume, in which the probability distribution along a depth direction represents the depth-estimation quality. The confidence value of a depth is computed by taking the probability sum over the four nearest-depth hypotheses on the probability volume [56]. Instead of estimating a single confidence map [56], we estimate multiple confidence maps based on the multistage cost volumes and integrate them into a representative confidence map. Let C_l be a confidence map corresponding to depth map D_l at stage $l \in \{1, \ldots, L\}$. For each stage except the last stage L, we upsample C_l and D_l to the same size as C_L and D_L. Each representative confidence value $\bar{C}(p)$ at pixel p is then calculated using the following weighted sum formula:

$$\bar{C}(p) = \frac{\sum_{l=1,\ldots,L} W_l(p) C_l(p)}{\sum_{l=1,\ldots,L} W_l(p)} \quad (2)$$

where $C_l(p)$ is the confidence value at pixel p and stage l and $W_l(p)$ its corresponding weight. The weight is defined by the relative depth difference $f_{rel-diff}(D_l, \bar{D}(p))$ between the corresponding depth $D_l(p)$ and the final depth $\bar{D}(p)$

$$W_l(p) = \exp\left(\frac{-f_{rel-diff}(D_l, \bar{D}(p))^2}{2\sigma_{wgt}^2}\right) \quad (3)$$

where σ_{wgt} is a constant value. We regard a pixel p with a confidence value $\bar{C}(p)$ lower than the threshold of 0.8 as an inconsistent depth and filter it out. Fig. 3 shows examples of the depth-filtering results.

B. Surfel Mapping

Surface model $\mathcal{F}$ represents a densely reconstructed surface of the target structures constructed by integrating the estimated depth maps using the surfel-based mapping method [14]. The mapping method represents a reconstructed surface as a set of surfels, where each surfel contains the following attributes: an xyz-position, normal, color, weight, radius, update time, and link to the attached keyframe. Given an estimated depth map D_t with its image I_t and camera pose $T_{w,t}$, the surfels for the current frame are initialized in the same way as in [14].

The initialized surfels are fused into the current surface model. The fusion method first associates each initialized surfel with

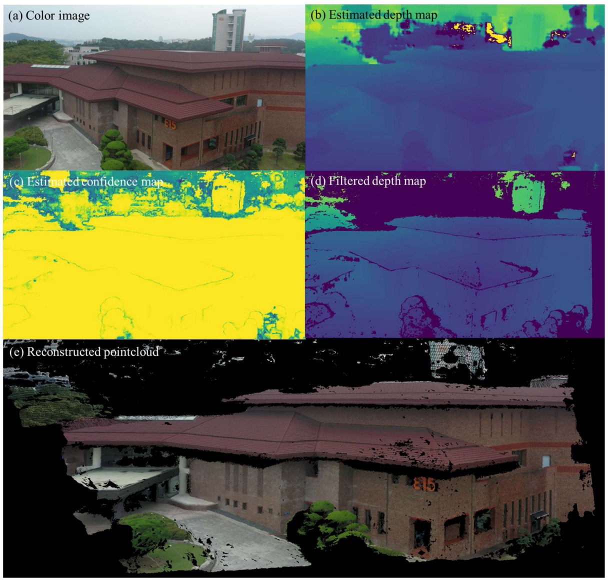


Fig. 3. Illustrations on depth estimation and filtering. (a) One reference image of Auditorium scene. (b) Estimated depth map from CasMVSNet [13]. (c) Confidence map. (d) Filtered depth map after photometric and geometric consistency checking. (e) Reconstructed point cloud.

the projected surfels in the surface model by rendering the surface model as an index map. Each pixel in the index map contains an index of a surfel in the surface model visible from the current camera frame. For each associated surfel pair, the method verifies the correspondence by checking if their depths and normals are similar. If the correspondence is valid, they are merged into the new surfel estimate using a weighted average of their attributes. The surfel weight is updated by summing their surfel weights. The rest of the initialized surfels are directly added to the surface model.

This surfel mapping method is the same as the online version of the conventional depth fusion in MVS [1], which fuses depths of all consistent views by averaging their 3-D positions and normals to suppress noise. The surfel mapping incrementally integrates depth maps from different frames to averaged point representations. We assume that the filtered depths are sufficiently reliable and have the same significance, similar to the conventional fusion approach [1]. So, we assign each initial surfel weight as a constant value (1.0 in this study). Therefore, the updated surfel weight represents how many times the surfel is reconstructed by the MVS. Similar to [14], our method labels the surfels that have not been updated within a certain period as inactive and filters out low-weight inactive surfels from the surface model. Therefore, the surface model is cleaned over time based on the surfel weights.

C. Processing Loop Closing

When loop closing occurs in the SLAM module, our system deforms the surface model and rebuilds the volumetric map according to the updated pose graph. Our method uses the nonrigid surfel deformation method [58] for the surface model. This method individually transforms the position and normal direction of each surfel to retain global consistency with the updated pose graph, instead of using a deformation graph [34]. Each surfel stores a link to the attached keyframe in the pose graph. For each surfel and its linked keyframe, the position and normal of the surfel are transformed according to the updated pose of the linked keyframe [58]. The surfels are updated in real time by parallel processing using a GPU.

This study does not consider the model-to-model registration that aims to minimize the point-to-plane error of the estimated depth and surface model as in the original surfel mapping method [14]. The registration approach could provide a more precise loop-closing result once; it does not preserve the global consistency of the surface model with the SLAM map-points. Therefore, we employ the surfel deformation method [58] that individually transforms the surfels based on the updated pose graph in the SLAM module to preserve the global consistency.

After deformation of the surface model, our method rebuilds the volumetric map. The method first reinitializes the entire area of the volumetric map to an unknown state and maps the deformed surfels to occupied volumes directly. It then extracts a list of free volume octree keys by casting rays from each updated pose into a view frustum. The ray-casting is performed at twice the coarseness of the resolution of $\mathcal{M}$ for fast updating. The keys that are already assigned to an occupied volume are rejected from the list. This avoids the situations where an occupied volume is incorrectly updated to a free volume. Finally, the extracted octree keys are updated to free volumes in batches.

V. Path Planning Method

To solve the problem of surface reconstruction and the exploration of an unknown environment, our method iteratively plans an exploration path by analyzing volumetric map $\mathcal{M}$ and surface model $\mathcal{F}$. Unlike the existing methods [22]–[30] that focus only on exploring a large unknown area, the proposed method explores the entire unknown area while simultaneously fully scanning the low-confidence surfaces to improve the performance of MVS reconstruction. As shown in Fig. 2(c), the newly proposed path-planning method is composed of the four stages. The first stage is global path planning (see Section V-A), in which a global path for efficiently exploring the remaining unknown area ξ_{global} is computed by analyzing the volumetric map. Our method iteratively computes global coverage of all frontiers and sequentially completes the entire model according to global coverage. A global path is determined as the first path segment of global coverage. In the second stage (see Section V-B), our method evaluates the reconstruction quality of the surface model and extracts low-confidence surfaces. The method then determines a set of target surface points $\bar{X}_{target}$ that needs to be scanned to achieve a high-quality surface model by clustering the low-confidence surfaces. The third stage is local inspection planning (see Section V-C), in which a local path ξ_{local}, which provides full coverage of the target surface points, is computed. The last stage is trajectory optimization (see Section V-D). This stage refines the local path to an optimized path that maximizes the performance of the MVS. Our method leverages heuristic information about the MVS for trajectory optimization.

Algorithm 1 shows the pseudocode of the proposed path-planning method. This algorithm represents an iterative step in a

Algorithm 1: Proposed Path Planning Algorithm.

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

/ Global path planning */*

1: $\{V_1, \ldots, V_N\} \leftarrow FrontierClustering(\mathcal{M})$

2: $\{Q_1, \ldots, Q_N\} \leftarrow GlobalSampling(\{V_1, \ldots, V_N\})$

3: $\{q_{\pi_1}, \ldots, q_{\pi_N}\} \leftarrow SolveGTSP(\{Q_1, \ldots, Q_N\}, q_{curr})$

4: $\{q_{NBV}, \xi_{global}\} \leftarrow GetPath(q_{curr}, q_{\pi_1})$

/ Local path planning */*

5: **while** $q_{curr} \neq q_{NBV}$ **do**

6:   **if** $TravelTime > \theta_{time}$ **then**

7:     $\bar{\mathcal{F}} \leftarrow PoissonSurfReconstruction(\mathcal{F})$

8:     $\bar{X}_{target} \leftarrow GetTargetSurfPoints(\bar{\mathcal{F}})$

9:     $\{\dot{Q}_1, \ldots, \dot{Q}_M\} \leftarrow LocalSampling(\bar{X}_{target}, \xi_{global})$

10:     $\{\dot{q}_{\pi_1}, \ldots, \dot{q}_{\pi_M}\} \leftarrow SolveGTSP(\{\dot{Q}_1, \ldots, \dot{Q}_M\}, q_{curr}, q_{NBV})$

11:     $\{Q_{ref}, \xi_{local}\} \leftarrow GetPath(q_{curr}, \{\dot{q}_{\pi_1}, \ldots, \dot{q}_{\pi_M}\}, q_{NBV})$

12:     $\xi^*_{local} \leftarrow OptmizePath(\bar{X}_{target}, Q_{ref}, \xi_{local})$

13:   **end if**

14:   $MoveToward(\xi^*_{local})$

15:   $Update(\mathcal{M}, \mathcal{F}, \xi_{global}, q_{curr})$

16: **end while**

loop; therefore, the MAV starts to plan a global path immediately after finishing the whole process of the algorithm. The method first computes an NBV configuration q_{NBV} and a global path ξ_{global} for sequentially exploring unknown areas in volumetric map $\mathcal{M}$ (lines 1–4). The method then plans local path ξ_{local}, which provides comprehensive scanning of local low-confidence surfaces in surface model $\mathcal{F}$ (lines 5–16). As mentioned previously, the local path is computed through three consecutive stages: target surface extraction, inspection path planning, and trajectory optimization. The local path is continually refined according to the updated local surfaces until the MAV reaches q_{NBV} (line 5). Using this approach, the MAV can rapidly explore the entire area while completely modeling the local surfaces of the target structure.

A. Global Path Planning

Our method determines the global path ξ_{global} by computing the global coverage path for the unexplored regions. The existing greedy methods [22]–[30], which iteratively move to the largest unknown area, sometimes generate inefficiently long trajectories by revisiting already explored areas. On the other hand, our method first obtains global coverage and sequentially explores the entire environment accordingly; therefore, this reduces the number of revisits to the same area and the total length of the exploration trajectory. Similar to this study, our previous methods [33], [43] decomposed an entire map into sectors and sequentially explored the sectors by computing the sector visitation order. However, the sector-based exploration method is not appropriate for our approach because the MVS can cover much wider range region than the decomposed sector. Therefore, instead of sector decomposition, the proposed method directly clusters the frontiers and sequentially explores each frontier cluster using the online MVS.

Our method first computes the global coverage of all frontiers to obtain a global path. Frontiers $V_{front} \subset V_{free}$ are determined as the set of free volumes V_{free} adjacent to the unknown volumes V_{unk} in $\mathcal{M}$. The method generates a set of frontier clusters $\{V_1, \ldots, V_N\}$ from V_{front} by greedily clustering the local frontiers in a specific range R_{front} (line 1). The global coverage path is the shortest path that explores each frontier cluster V_i once, starting from the current configuration q_{curr}.

To compute a global coverage path, our method first generates a set of configuration samples to ensure that each frontier cluster is observed and then determines the coverage path from the sample set. For each frontier cluster V_i, the method computes its centroid position $\bar{c}_i$ and estimates the normal direction $\bar{n}_i$ based on local least square fitting. The method generates a set of feasible robot configurations $Q_i \subset Q$ that obtain views of V_i by the dual sampling method [51], [59] (line 2). The dual sampling method determines a cover region Ω_i, in which target cluster V_i is visible, and then generates configuration samples in Ω_i. The cover region is determined by inversely composing a view frustum from the center $\bar{c}_i$ to its normal direction $\bar{n}_i$ and checking the visibility by casting a ray from $\bar{c}_i$ into the view frustum. The method generates a set of uniform samples Q_i in Ω_i and checks whether each sample $q \in Q_i$ can observe the frontiers in V_i. As some frontiers cannot be observed from any configuration in Ω_i, our method checks if sample q satisfies a certain percentage of coverage for V_i (80% in this study) rather than full coverage. Samples that do not provide frontier coverage are eliminated from Q_i.

After dual sampling, the method computes a coverage path of the frontier clusters $\{V_1, \ldots, V_N\}$ by finding the minimum distance trajectory that visits at least one sample $q_i \in Q_i$ from each sample set Q_i. This problem is formulated as a solution to the *generalized traveling salesman problem* (GTSP). Given a set of sample sets $\{Q_1, \ldots, Q_N\}$, the GTSP solution determines the minimum-cost path starting from a current configuration q_{curr} that visits each sample q_i in each sample set Q_i exactly once (line 3). The determined optimal path is represented as a sequence of selected samples $\{q_{\pi_1}, \ldots, q_{\pi_N}\}$, where $\Pi = \{\pi_1, \ldots, \pi_N\}$ is the permutation of the sample set indices $\{1, \ldots, N\}$ representing the visitation sequence. To solve the GTSP, the generalized 2-opt neighborhood approach [60] is used.

The cost of each configuration pair is defined as the Euclidean distance of their connected path. To efficiently compute the path in the whole 3-D space, we use an approximate representation of the environment by composing uniform grid cells from specific depth nodes on a volumetric map $\mathcal{M}$. We construct a weighted, undirected adjacency graph G_{adj}, where its vertices represent the center positions of the uniform grid cells in free space, and its edges are composed of adjacency connections with the cost of Euclidean distance. A path of each configuration pair is computed on the graph G_{adj} using the A* planner. We define the heuristic function of the A* planner as the Euclidean distance from a vertex to a goal.

Finally, the method determines an NBV q_{NBV} as the first sample q_{π_1} to sequentially visit the computed coverage path and computes the global path ξ_{global} from q_{curr} to q_{NBV} (line 4). It first initializes the path ξ_{global} from the computed path on G_{adj} and then computes the shortest piecewise-linear path using the path shorten method, as in [61].

B. Target Surface Extraction

This section describes a method for extracting the scanning targets that must be refined for high-quality surface modeling. The scanning targets are determined by evaluating the reconstruction quality of the surface model. The previous methods [10], [44] typically predicted the reconstruction quality of a tentative coarse model by using a simple coverage information. However, the proposed method more precisely evaluates the reconstruction quality because the online reconstruction model is denser than the tentative coarse model.

Our method first predicts a tentative 3-D model $\bar{\mathcal{F}}$ from the surfels in surface model $\mathcal{F}$ and then evaluates the surface quality of the tentative model (line 7). It applies the screened Poisson reconstruction [62] to predict the tentative model. For fast computation, the method sets the maximum depth parameter of the reconstruction algorithm to eight. This study predicts the whole model instead of considering a subregion as in [43] because the MVS can cover a wide range of regions. However, it will be possible to reduce the computation time for generating a tentative model by focusing only on a subregion around the global path.

Our method generally obtains the high-quality results of Poisson reconstruction because the artifacts on $\mathcal{F}$ are thoroughly removed by applying several filtering steps, including photometric and geometric consistency checks and surfel-weight-based filtering. Furthermore, to provide a smoother and noise-reduced reconstruction, we set the parameter of minimum sample number in the Poisson reconstruction algorithm to 15, which is a large value. The tentative model represents clean isosurfaces of the raw surfels, from which the surface tendencies of the unscanned area can be inferred. The reconstructed surfaces in $\bar{\mathcal{F}}$ are represented as a set of oriented points X, referred to as surface points, where each point $x_k \in X$ contains a 3-D position and normal values.

For each surface point x_k, our method measures the confidence value that represents the reconstruction quality in location x_k. We present a new confidence measure that considers both density and weights of the neighboring surfels. The surfel weight represents the reliability of the corresponding surfel, and a high surfel weight indicates that the surfel has been updated significantly. Our method first finds the set of neighboring surfel points $\mathcal{N}(x_k)$ by the *K*-nearest neighborhood of x_k ($K = 20$ in this study). It then calculates the confidence of surface point x_k using the following function:

$$f_{conf}(x_k) = \frac{\sum_{x_n \in \mathcal{N}(x_k)} f_{dist}(x_k, x_n)\bar{w}_k}{\#(\mathcal{N}(x_k))} \quad (4)$$

where $\#(\mathcal{N}(x_k))$ is the number of neighboring surfels and $\bar{w}_k$ is a normalized surfel weight that ranges from zero to one. It is defined as $\bar{w}_k = \min(w_k/w_{max}, 1.0)$, where w_k is a surfel weight of x_k, and w_{max} is a user-defined maximum weight ($w_{max} = 3.0$ in this study). $f_{dist}(x_k, x_n)$ is the spatial weighting function penalizing long distances between x_k and x_n; it is defined as

$$f_{dist}(x_k, x_n) = \exp\left(\frac{-\|x_k - x_n\|^2}{2\sigma_{dist}^2}\right) \quad (5)$$

where $\|\cdot\|$ is the l_2-norm in 3-D space (xyz-coordinates) and σ_{dist} is a constant value. This function provides low confidence to the surface points of sparsely reconstructed regions, so it reflects the density of the neighboring surfels as well.

Our method determines the low-confidence surface points by extracting the surface points whose confidences are lower than 0.2. The surface points far from the surfels in $\mathcal{F}$ are rejected from the set of low-confidence surfaces because an empty region with no surfels around is sometimes reconstructed inaccurately. The method then groups adjacent low-confidence surface points using greedy Euclidean clustering. The clustering method selects a surface point at random and clusters its neighboring points within a distance range of R_{surf} and an angle range of R_{angle} between their normals. This grouping process is repeated until every low-confidence surface point is assigned to a cluster. Clusters with few surface points are removed. Finally, our method determines a set of target surfaces $\bar{X}_{target}$ by computing the averaged surface point $\bar{x} \in \bar{X}_{target}$ for each cluster (line 8). Fig. 4 shows an example of the extraction process of the target surface points.

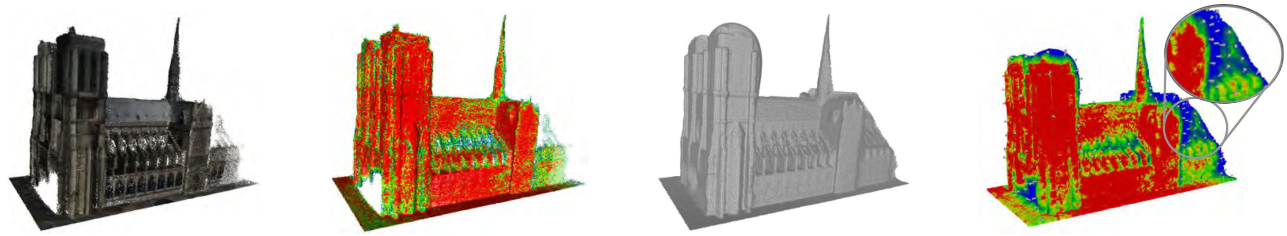
C. Local Inspection Path Planning

This section describes the planning method of a local inspection path ξ_{local} that provides a full visual coverage of the target surface points $\bar{X}_{target}$ in the shortest distance. Similar to the global coverage planning in Section V-A, our method uses dual sampling and the GTSP algorithm to plan an inspection path. For each target surface point $\bar{x}_j \in \bar{X}_{target}$, our method performs dual sampling to generate a set of sample configurations $\dot{Q}_j \subset Q$, where each sample $\dot{q} \in \dot{Q}_j$ observes target point $\bar{x}_j$ (line 9). It inversely composes a view frustum from $\bar{x}_j$ to its normal direction and generates uniform samples $\dot{Q}_j$ in the view frustum by checking the visibility of $\bar{x}_j$. The incidence angle is additionally considered for a visibility check; the angle between the surface normal and view direction must be smaller than the minimum incidence angle of 60°. To prevent the computed path from becoming significantly longer than ξ_{global}, we restrict the sampling space $Q_{sample} \subset Q$ as

$$Q_{sample} = \{q \in Q | \|q_{curr} - q\| + \|q - q_{NBV}\| \leq \gamma d_{global}\} \quad (6)$$

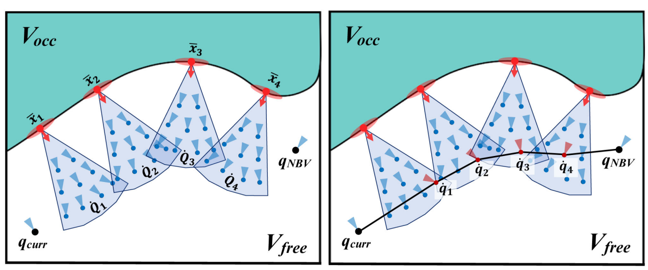
where d_{global} is the path length of a global path ξ_{global} and γ is a constant value ($\gamma = 1.3$ in this study). The method rejects a sample q_j outside the sampling space Q_{sample} from each sample set $\dot{Q}_j$. A target surface point with an empty sample set is excluded from $\bar{X}_{target}$.

Given a set of configuration sets $\{\dot{Q}_1, \ldots, \dot{Q}_M\}$, the GTSP algorithm is also used to determine the minimum cost tour (line 10). The tour starts from q_{curr}, visits a sample $\dot{q}_j$ in each sample set $\dot{Q}_j$ exactly once, and ends at q_{NBV}. The cost of two



(a) (b) (c) (d)

Fig. 4. Extraction process of target surface points: Our method first obtains (a) reconstructed surfel points and (b) their surfel weight values from surface model $\mathcal{F}$. The high-weight surfel points are shown in red and low weights in blue. The method then predicts (c) tentative 3D model $\bar{\mathcal{F}}$ from the surfel points using the screened Poisson reconstruction algorithm. The method measures the confidence of each reconstructed surface point in $\bar{\mathcal{F}}$. Finally, (d) the target surface points, gray normal points, are determined by extracting the low confidence surfaces. The target surfaces on the sparse point region (zoomed-in local region) can be estimated from the surface trend of $\bar{\mathcal{F}}$.



(a) (b)

Fig. 5. Illustration of the local inspection path planning. The global path planning is also similar to this process except for the end configuration q_{NBV}. The process is depicted in 2-D for clarity. (a) Dual sampling method first generates a set of coverage samples $\dot{Q}_j$ for each target surface point $\bar{x}_j \in \bar{X}_{target}$ by inversely composing a view frustum from $\bar{x}_j$ to its normal direction. (b) GTSP algorithm determines the minimum-cost path that starts from a current configuration q_{curr}, visits a sample $\dot{q}_j$ in each sample set $\dot{Q}_j$ exactly once, and ends at q_{NBV}.

connecting configurations q_1 and q_2 is defined as the motion execution time [50]:

$$f_{cost}(q_1, q_2) = \max\left(\frac{\|q_1 - q_2\|}{v_{max}}, \frac{|\psi_1 - \psi_2|}{\dot{\psi}_{max}}\right) \tag{7}$$

The optimal tour is represented as a sequence of samples $\{\dot{q}_{\pi_1}, \ldots, \dot{q}_{\pi_M}\}$, where $\Pi = \{\pi_1, \ldots, \pi_M\}$ is the permutation of the sample set indices $\{1, \ldots, M\}$. Fig. 5 illustrates the dual sampling and the GTSP results.

Finally, our method determines the local inspection path ξ_{local} by directly connecting consecutive configurations in $\{\dot{q}_{\pi_1}, \ldots, \dot{q}_{\pi_M}\}$ from q_{curr} to q_{NBV}. Unlike global path planning (see Section V-A), local planning computes a path from configurations sampled in a continuous space of a restricted local area. The RRT* planner [63], a sampling-based approach, effectively computes the local paths in a continuous space. Collisions rarely occur on the local path planning because the local paths mostly compute on open free spaces. Therefore, the method first checks the direct connection and then uses the RRT* planner (with path shorten as in [61]) if a collision occurs. The set $\{\dot{q}_{\pi_1}, \ldots, \dot{q}_{\pi_M}\}$ is referred to as the reference configuration set Q_{ref}, where the configurations are used as reference views for reconstructing their target surfaces $\bar{X}_{target}$.

D. Trajectory Optimization

Even if the target surface points are fully scanned by the planned inspection path ξ_{local}, the MVS method is not guaranteed to produce the accurate reconstruction. Therefore, in order to construct the target surfaces more accurately, the inspection path must be refined. This section introduces a new measure to predict reconstruction quality using the MVS heuristic information. By applying the measure, our method optimizes the local inspection path to enhance the accuracy of target surface reconstruction. We formulate the path optimization problem as an *informative path planning problem*. By solving the problem, we can obtain a polynomial trajectory that increases the MVS performance in a short travel time. Fig. 6 shows the examples of the updated optimal trajectories with global and local paths.

The following sections introduce a score function for predicting the MVS reconstruction quality and describe how to apply it to the trajectory optimization problem.

1) Reconstruction Quality Prediction: We describe how to incorporate heuristic information into the stereo-pair selection for high-quality MVS reconstruction. Many studies [3], [11], [47], [48], [64] have demonstrated that the reconstruction quality of a surface point depends on several geometric factors, including triangulation angles, relative image resolutions, and focus angles. Given a stereo-pair of reference view configuration q_{ref} and source view configuration q_{src}, we define a new score function that predicts the reconstruction quality of a target surface point $\bar{x}$ as

$$f_{src}(q_{src}, q_{ref}, \bar{x}) = f_{vis} \cdot f_{prx} \cdot f_{rd} \cdot f_{foc} \tag{8}$$

where f_{vis} is a visibility function that returns the value 1 if q_{src} obtains a view of $\bar{x}$, and 0 otherwise. f_{prx}, f_{rd}, and f_{foc} are the score functions that are related to MVS heuristics: *parallax*, *relative distance*, and *focus*, respectively. The following describes each score function in detail:

Parallax: According to the parallax angle of a stereo pair, there is a tradeoff between triangulation accuracy and matchability [3], [11], [55]. Wide-baseline measurements with a large parallax angle can increase triangulation accuracy. On the other hand, the success probability of stereo matching in the stereo search along the long epipolar line decreases for a long baseline. Given a parallax angle α between the stereo pair of q_{ref} and q_{src},

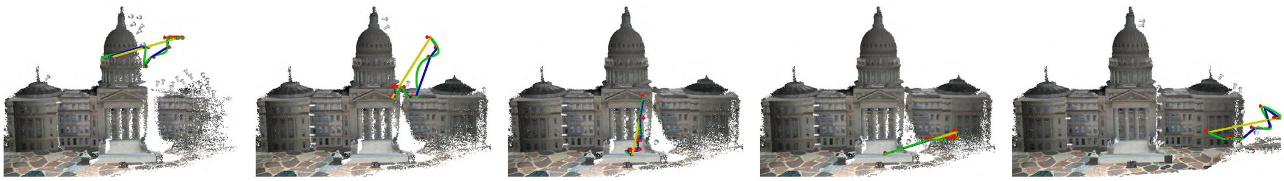


Fig. 6. Examples of the global path (yellow), local inspection path (blue), and optimized path (green). The paths were iteratively computed based on the incrementally updated models from (a) to (e). The gray and red arrows represent the target surface points and their reference view configurations, respectively.

score function f_{prx}, which represents the informativeness of α for reconstructing the correct surface, is defined as

$$f_{prx}(\alpha) = \exp\left(-\frac{(\alpha-\alpha_0)^2}{2\sigma_{prx}^2}\right) \tag{9}$$

where σ_{prx} is a constant value, and α_0 is the desired parallax angle, which is heuristically determined as 11°.

Relative distance: The source image should have a resolution similar to that of the reference image for accurate stereo matching [3]. We assume that similar resolution images of a specific surface can be obtained by views at the same distance from the surface. Let $dist_{ref}$ and $dist_{src}$ be the distance between $\bar{x}$ and reference view q_{ref}, and the distance between $\bar{x}$ and the source view q_{src}, respectively. Score function f_{rd} for the relative distance between $dist_{ref}$ and $dist_{src}$ is defined as

$$f_{rd}(dist_{src}, dist_{ref}) = \frac{\min(dist_{src}, dist_{ref})}{\max(dist_{src}, dist_{ref})}. \tag{10}$$

If $dist_{ref}$ and $dist_{src}$ are similar, f_{rd} has a score close to 1.

Focus: It is preferable that the target surface region be projected around the principal point of the source image to reduce the reprojection error in triangulation [47], [48]. Let $r_{\overline{co}}$ and $r_{\overline{cx}}$ be rays from camera center c of a source image to principal point o and to surface point $\bar{x}$, respectively. Given focus angle β between the rays $r_{\overline{co}}$ and $r_{\overline{cx}}$, the score function f_{foc} penalizing a large β is defined as

$$f_{foc}(\beta) = \exp\left(-\frac{\beta^2}{2\sigma_{foc}^2}\right) \tag{11}$$

where σ_{foc} is a constant value.

2) Trajectory Optimization: After planning an inspection path ξ_{local} for the target surfaces $\bar{X}_{target}$, our method refines the path to maximize MVS performance (line 12). Our method computes a path that visits each reference configuration $q_r \in Q_{ref}$ while simultaneously improving the reconstruction quality of each target surface $\bar{x}_r \in \bar{X}_{target}$. Let $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_M\}$ be a set of disjoint path segments, where each segment ξ_s is a path connecting consecutive reference configurations. The aim is to find an optimal set of path segments $\boldsymbol{\xi}^*$ that maximizes the performance of MVS about the target surfaces while meeting a budget constraint. We formulate the problem as an informative path-planning problem, which finds the most informative path passing predefined waypoints within the given budget constraint. Let $\mathcal{I}(\xi_s)$ be a utility function that returns the information quality gathered along ξ_s and $TIME(\xi_s)$ be the corresponding travel time. The most informative path is computed by solving the following optimization problem:

$$\boldsymbol{\xi}^* = \text{argmax}_{\boldsymbol{\xi}} \sum_{\xi_s \in \boldsymbol{\xi}} \frac{\mathcal{I}(\xi_s)}{TIME(\xi_s)}$$

$$\text{s.t } TIME(\xi_s) \le B_s \text{ for every segment } s \tag{12}$$

where B_s is the time budget of segment s. The budget is defined as $B_s = \gamma' \times TIME(\bar{\xi}_s)$, where γ' is a constant value of 1.3, and $\bar{\xi}_s$ is the shortest path from the starting configuration to the end configuration of ξ_s. Each $\bar{\xi}_s$ is determined from the inspection path computed in Section V-C.

We define the informativeness of a path segment ξ_s as the reconstruction quality of target surfaces $\bar{X}_{target}$ when performing the MVS with the view configurations on ξ_s. Given sequential reference configurations $Q_{ref} = \{q_1, \ldots, q_M\}$ and their target surfaces $\bar{X}_{target} = \{\bar{x}_1, \ldots, \bar{x}_M\}$, we assume that each reference configuration q_r only participates in reconstructing its target surface $\bar{x}_r$. A current path segment ξ_s is not involved in reconstructing target surfaces $\{\bar{x}_1, \ldots, \bar{x}_{s-1}\}$ of the previous reference configurations $\{q_1, \ldots, q_{s-1}\}$; therefore, it considers only a subset of reference configurations $\bar{Q}_{ref} = \{q_s, \ldots, q_M\}$ and their target surfaces $\{\bar{x}_s, \ldots, \bar{x}_M\}$. Our method extracts a sample set of discrete configurations Q_s at specific time intervals along ξ_s. For each pair of a sample configuration $q_i \in Q_s$ and a reference configuration $q_r \in \bar{Q}_{ref}$, the method predicts the reconstruction quality of its target surface $\bar{x}_r \in \bar{X}_{target}$ using the heuristic function $f_{src}(q_i, q_r, \bar{x}_r)$ (8) and accumulates them to measure the informativeness. The utility function $\mathcal{I}$ for a path segment ξ_s is defined as

$$\mathcal{I}(\xi_s) = \sum_{q_i \in Q_s} \sum_{q_r \in \bar{Q}_{ref}} f_{src}(q_i, q_r, \bar{x}_r). \tag{13}$$

This function quantifies the reconstruction quality of the target surfaces by accumulating the MVS heuristic scores of discrete configurations taken along ξ_s.

We use the online informative path-planning algorithm [65] to solve the optimization problem (12). The algorithm first determines global waypoints by greedily selecting the most informative viewpoints. It then finds an optimal continuous trajectory that passes the waypoints with maximum informativeness using an evolutionary strategy [66]. Our method sets the reference configurations Q_{ref} as the global waypoints and computes the most informative trajectory using the evolutionary strategy, similar to [65].

As described in [65], our method computes a polynomial trajectory [67] that a MAV can dynamically follow each waypoint of the local path ξ_{local}. The polynomial trajectory ensures a smooth motion without requiring step inputs to the MAV's actuators. It also provides continuous motion at the shared configuration between segments. We use a polynomial segment to connect two waypoints of a ξ_s and evaluate the informativeness of the segment using $\mathcal{I}(\xi_s)$.

An optimal polynomial trajectory is obtained by the *covariance matrix adaptation evolution strategy* (CMA-ES) [66]. The CMA-ES is based on evolutionary algorithms, which work equally well for both nonlinear and nonconvex problems in continuous space. The CMA-ES first generates sample trajectories according to a multivariate Gaussian distribution and evaluates the utility for each sample trajectory using the utility function $\mathcal{I}$. It then selects the best candidate solutions and updates the parameters of the distribution from the selected solutions. The CMA-ES iterates this optimization process until the solutions converge.

E. Reference and Source View Selection

The camera moves along the computed optimal path ξ^*_{local} while storing the image frames in the SLAM database. Our method consistently determines the reference images every time a keyframe is extracted. The keyframes are consistently extracted at regular frame intervals. Our view selection method extracts neighboring keyframes using the covisibility graph in the SLAM module and then selects the source images from among them. When the camera moves along a path segment $\xi_s \subset \xi^*_{local}$, our method focuses on reconstructing the corresponding target surface $x_s \in \bar{X}_{target}$. Given a reference image with its target surface, the method selects five keyframes among the neighboring keyframes as source images by evaluating the score function. The NBV configuration q_{NBV} at the end of ξ^*_{local} does not have a target surface point; thus, it selects the five keyframes that share most map-point observations as source images.

VI. Experimental Results

We conducted simulations and real-world experiments to evaluate the performance of the proposed method. In the simulation experiments, the modeling performance of the proposed method was quantitatively compared with the performance of existing view-planning methods. Because ground-truth information can be used in simulation environments, it is possible to make a reliable quantitative comparison of the modeling performance. On the other hand, in real-world experiments, it is difficult to evaluate results quantitatively because ground-truth information is not available for real-world scenes. Therefore, we considered a qualitative evaluation and feasibility demonstration of our method in real-world applications, rather than a quantitative evaluation.

A. Simulation Experiments

We conducted the simulation experiments in the RotorS simulation environment [68] and used a Firefly hexacopter MAV model as the scanning platform. The MAV was equipped with a forward-looking stereo camera, which had a field of view $[60^\circ, 90^\circ]$ and a pitch angle of 15° downward. The camera captured image frames with a resolution of 752×480. A stereo version of ORB-SLAM [12] was used to estimate the camera poses and map points on a metric scale. For reliable pose estimation, we composed feature-rich environments by covering the background with textured scenes. We also restricted the maximum translational speed to 0.5 m/s and rotational speed to 0.25 rad/s. We used only the left images of the stereo camera to process the MVS reconstruction.

We considered four target infrastructures:[1] *Alexander Nevsky Cathedral* (scenario 1), *State Capitol* (scenario 2), *Castle* (scenario 3), and *Notre Dame Cathedral* (scenario 4). The structures in scenarios 1 and 2 are composed of relatively fewer textured surfaces and include dome-shaped substructures. The structures in scenarios 3 and 4 are composed of highly textured surfaces and complex subregions. In particular, the structure in scenario 4 contains many complex parts in which occlusion occurred frequently during scanning.

We separately demonstrated the effectiveness of each planning stage (global path planning $\mathbf{G}$, local inspection path planning $\mathbf{L}$, and trajectory optimization $\mathbf{O}$) in our method by conducting the following ablation experiments:

- $\mathbf{Ours_G}$: Only GPP, as described in Section V-A. This produces an exploration path that simply follows the global coverage of all frontiers.
- $\mathbf{Ours_{G+L}}$: Local inspection path planning with GPP as described in Section V-C. This plans an inspection path that covers low-confidence surface points.
- $\mathbf{Ours_{G+L+O}}$: Path-planning method including all stages. This contains the trajectory optimization discussed in Section V-D and view-selection methods from Section V-E.

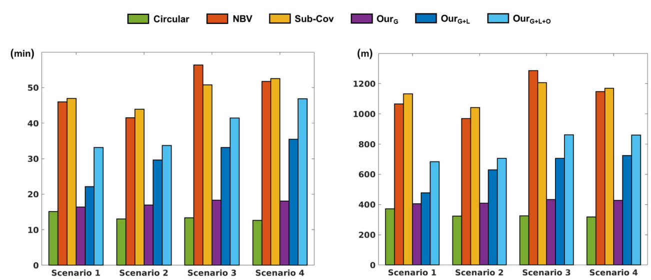
The proposed methods were compared with explore-then-exploit methods [8], [44]. The explore-then-exploit methods first construct an initial coarse model from a default trajectory and then compute the rescanning paths by analyzing the coarse model. We considered two state-of-the-art explore-then-exploit methods:

- **NBV** [44]: This method iteratively determines the best viewpoint online by evaluating the surface coverage. The determined viewpoint observes the largest uncovered area based on partial reconstruction. The method first generates a set of sampled surface points, iso-points, by performing Poisson reconstruction and Poisson disk sampling. The method uniformly samples the viewpoints within a restricted range (20 m) around the current location to improve the path efficiency. For each viewpoint, it evaluates the uncovered area by accumulating the projection ratio of uncovered iso-points. It then determines an NBV and computes the path to the NBV. Like our local path-planning, the method first checks the straight-line path and then uses the RRT* planner if a collision occurs.
- Submodular coverage (Sub-Cov) [8]: This method computes the coverage path of an initial coarse model by solving a submodular orienteering problem. Similar to our global path-planning, it first generates an adjacency graph

[1] [Online]. Available: http://3dwarehouse.sketchup.com/

TABLE II
COMPARISON OF MODELING QUALITY FOR DIFFERENT PATH-PLANNING METHODS ON THE FOUR SIMULATION SCENARIOS

	Scenario 1			**Scenario 2**			**Scenario 3**			**Scenario 4**		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Circular	0.6678	0.7310	0.6964	0.7895	0.6654	0.7215	0.7803	0.5513	0.6449	0.7823	0.5357	0.6361
NBV [44]	0.7718	0.8855	0.8236	0.8613	0.7926	0.8234	0.8446	0.7862	0.8152	0.7982	0.7508	0.7746
Sub-Cov [8]	0.7834	0.9116	0.8432	0.8724	0.8126	0.8427	0.8346	**0.8372**	0.8364	0.8102	0.7808	0.7957
Ours_{G}	0.7619	0.8448	0.8015	0.8247	0.7202	0.7692	0.8302	0.7551	0.7927	0.7843	0.6902	0.7349
$\text{Ours}_{\text{G+L}}$	0.7793	0.9048	0.8374	0.8367	0.8038	0.8219	0.8388	0.7902	0.8151	0.7907	0.7922	0.7927
$\text{Ours}_{\text{G+L+O}}$	**0.7998**	**0.9238**	**0.8571**	**0.8781**	**0.8206**	**0.8550**	**0.8571**	0.8213	**0.8398**	**0.8213**	**0.8308**	**0.8274**



(a) (b)

Fig. 7. Path efficiency results of the simulation scenarios comparing different path-planning methods on the four simulation scenarios. The results of NBV and Sub-Cov show the cumulative time and path length during the whole explore-then-exploit process.

representing the entire environment and then computes a coverage path on the graph. The method finds a path that observes the target surfaces from diverse viewing angles as much as possible in the graph. Each path of a configuration pair is computed from the adjacency graph using the A* planner.

To reconstruct a coarse model, we performed an initial scan with a circular trajectory around the target space at a camera pitch of 25°. To achieve a detailed reconstruction at a specific resolution, we restricted the maximum scanning range to 50 m for path planning. We used a circular trajectory as the path to scan the largest target area with the maximum scanning range. Both the initial and final 3-D models were constructed using the proposed online modeling system in Section IV. NBV and Sub-Cov do not extract target surface points; thus, they use the neighbor frames that share most map points as source images. We set the total travel budgets of Sub-Cov and NBV as the averaged path distance result from $\text{Ours}_{\text{G+L+O}}$.

The performances were evaluated based on two perspectives: path efficiency and modeling quality. To evaluate the path efficiency, we computed the *completion time* and *path length*. The modeling quality refers to the evaluation process and metrics described by Knapitsch *et al.* [69]. We first aligned the reconstructed point cloud to the ground-truth model by registering the estimated camera poses to ground-truth camera poses. We then performed iterative closest point registration to refine the alignment of the reconstructed point cloud. After the alignment, we resampled the reconstructed and ground-truth point clouds on a voxel grid with a voxel size of 0.05 m. Both point clouds were compared using *precision* and *recall*. Precision is defined as the percentage of reconstructed points that are close to a ground-truth point, and recall is defined as the percentage of ground-truth points that are close to a reconstructed point. The close points are determined by a distance threshold τ, which was set to 0.1 m in these experiments. Both precision and recall can be captured in a representative metric: the $F - score = \frac{2(precision \times recall)}{precision + recall}$. Fig. 7 and Table II present the experimental results for the path efficiency and modeling quality averaged over five executions, respectively. Fig. 8 depicts the reconstructed models and MAV trajectories of the best trial.

Our methods had better path efficiency performance in terms of completion time and path length compared to the other methods because our methods do not require an initial scan. As compared to Sub-Cov (with initial scan), $\text{Ours}_{\text{G+L+O}}$ reduced the average completion time by 20.42% and the average path length by 31.75%. As shown in Fig. 8, NBV generated complex trajectories that frequently overlapped because it focused only on planning the local path. The paths of Sub-Cov sometimes revisited routes that had already been passed in an earlier trajectory. On the other hand, our method reduced the number of revisits by considering the global coverage sequence and completely scanning the local regions. Thus, our method can achieve improved path efficiency performance without prior model information.

$\text{Ours}_{\text{G+L+O}}$ achieved the best modeling performance in terms of F-score. The F-scores of NBV were the poorest compared to Sub-Cov and $\text{Ours}_{\text{G+L+O}}$ in all scenarios. NBV focuses only on scanning the largest uncovered surface area while disregarding minor surfaces, so its reconstructed models may be incomplete. In scenarios 1 and 2, the performance gaps of the recalls between Sub-Cov and $\text{Ours}_{\text{G+L+O}}$ are not significant because occlusion rarely occurred during scanning. However, the precision of $\text{Ours}_{\text{G+L+O}}$ was much higher than those of Sub-Cov; this indicates that the path considering the MVS heuristics can increase the reconstruction accuracy. In scenario 4, occlusions of complex parts occurred frequently during scanning, so a complete and accurate modeling of the scene was not guaranteed. Therefore, most methods have low precision and recall performances, but $\text{Ours}_{\text{G+L+O}}$ showed superior performance. $\text{Ours}_{\text{G+L+O}}$ focuses

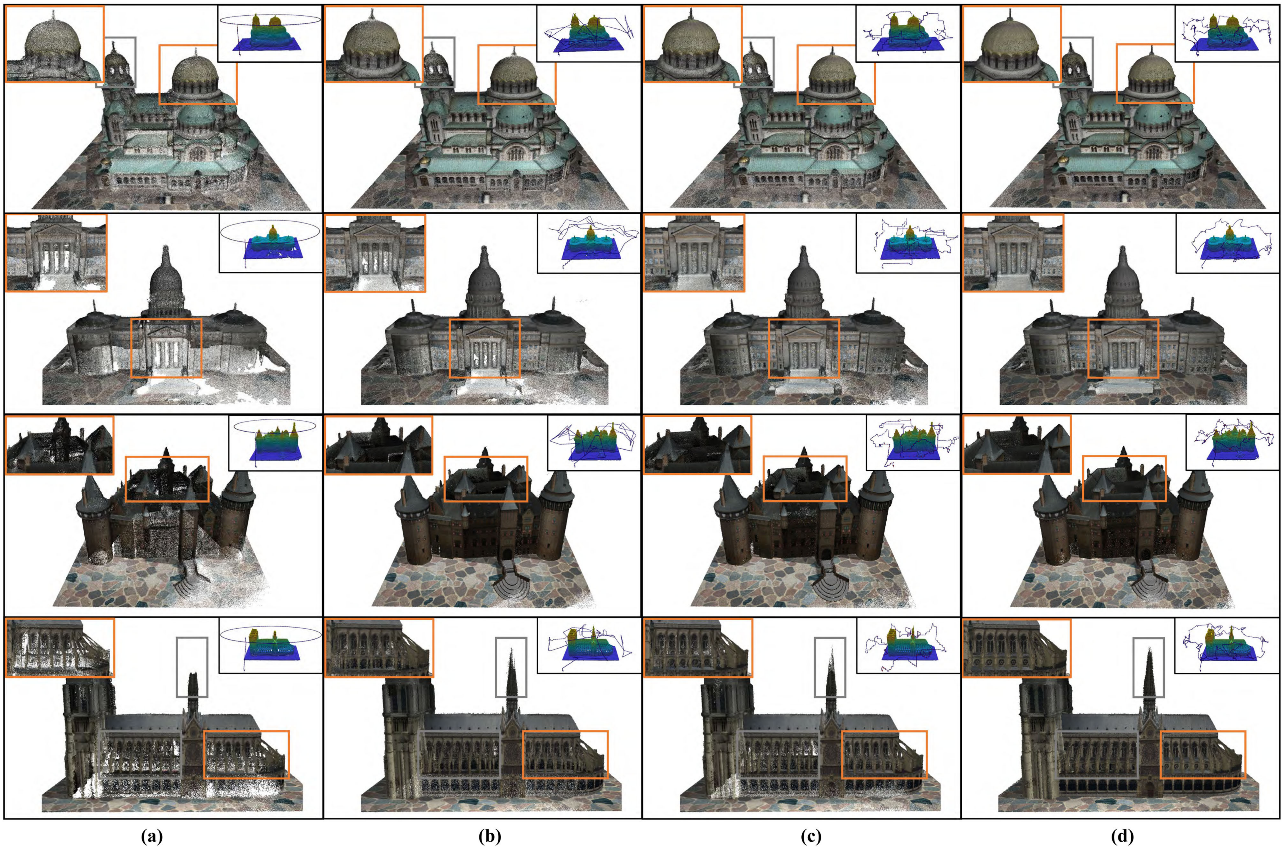


Fig. 8. Reconstructed 3-D models and volumetric maps with trajectories taken by the MAV (first row: Scenario 1, second row: Scenario 2, third row: Scenario 3, and fourth row: Scenario 4). For each 3-D model, we provide a zoomed-in detail view for the region highlighted with orange boxes. Our method was able to obtain denser and more accurate point clouds that made more complete models, compared to other methods.

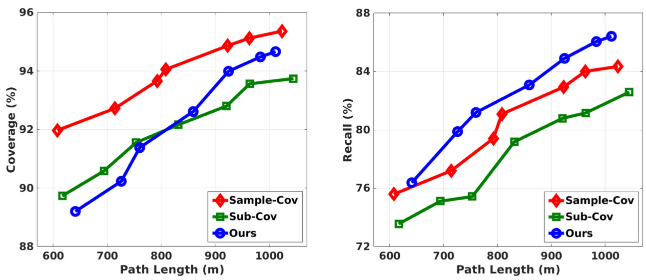
on completing the insufficiently reconstructed regions by examining the completeness of the reconstructed surfaces. This approach enhances the modeling performance even in complex scenes, such as in scenario 4.

Ours_{G} completed the modeling in the shortest time but had the lowest modeling performance. $\text{Ours}_{\text{G+L}}$ generally has better modeling performance than NBV because it covers poorly reconstructed local surfaces through multiple viewpoints instead of a single viewpoint. As compared to $\text{Ours}_{\text{G+L}}$, $\text{Ours}_{\text{G+L+O}}$ increased the average F-score by 0.028. This suggests that the proposed trajectory optimization method is effective for MVS reconstruction.

B. Evaluation of Surface Coverage

In this experiment, we evaluated the performance of our method in terms of the surface coverage and investigated the relationship between the surface coverage and the model completeness. The surface coverage refers to the percentage of the observable surfaces given the entire surfaces of the ground-truth 3-D model. The observable surfaces were estimated by evaluating the viewpoints sampled at regular time intervals from a planned path. The coverage performance of our method was compared with that of Sub-Cov and the sampling-based coverage method (Sample-Cov) [70]. Sample-Cov plans a coverage path of a target 3-D model offline by using a sampling-based approach. Sample-Cov first generates a set of coverage samples by dual-sampling [71] in a continuous space and then computes their connecting path by solving the multigoal planning problem using the RRT* planner. Sample-Cov guarantees the property of probabilistic completeness, meaning the algorithm will eventually find a feasible solution if it exists. Furthermore, it significantly reduces the total path length by the asymptotically optimal local improvements. We considered the coverage path of Sample-Cov as a solution close to optimal and compared it with the result of our method.

Fig. 9 shows the coverage and modeling performance of each method under different path lengths in scenario 4. The model completeness was evaluated by measuring the recall of a reconstructed model. Sub-Cov and Sample-Cov generated several coverage paths with different path lengths by changing the time budget and the number of coverage samples, respectively. Their coverage paths were computed from the ground-truth 3-D model instead of a coarse model. The performances of our method ($\text{Ours}_{\text{G+L+O}}$) were measured using the final exploration trajectory that was planned online without the ground-truth information. We also generated the exploration paths with different path lengths by increasing the parameter γ from 1.15 to 1.45. The minimum number of observations for classification of covered surface was set as three.

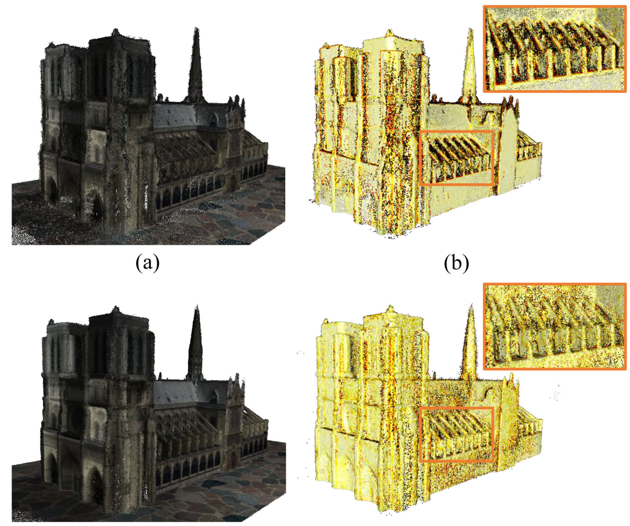


(a) (b)

Fig. 9. Performances of (a) surface coverage and (b) model completeness (recall) under different path lengths using Sub-Cov [8], Sample-Cov [70], and ours in scenario 4. Both Sub-Cov and Sample-Cov computed the different coverage paths for the ground-truth 3-D model by changing the time budget and the number of coverage samples, respectively. Our method ($\text{Ours}_{\text{G+L+O}}$) planned the different exploration paths by changing the parameter γ.

As seen in Fig. 9(a), Sample-Cov achieved the best coverage performances. Sub-Cov plans a coverage path from restricted samples in a discretized environment; therefore, it always had lower coverage performances than Sample-Cov. Our method had better coverage performances than Sub-Cov in the paths longer than 860 m while it had worse performances in the shorter paths. The local inspection planning in our method also plans a coverage path using a sampling-based approach similar to Sample-Cov. If there is no limitation of the sampling area, the proposed method can eventually find the complete coverage solution as well. However, some target surfaces cannot always be covered given the sampling area limited by γ. The higher value of γ, the more target surfaces can be covered. Therefore, our method has significantly better coverage performance as the path length increases.

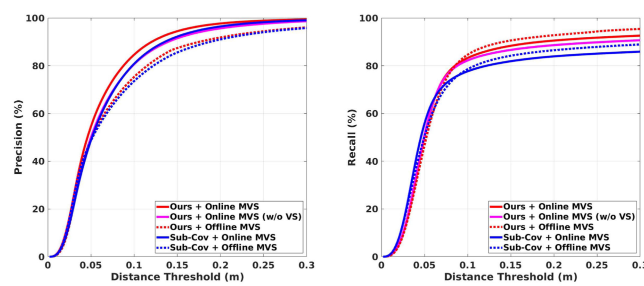
Fig. 9(b) shows the modeling performances with respect to the model completeness. As can be seen in the figure, our method had better modeling performances than Sample-Cov. This indicates that a high surface coverage does not always guarantee better MVS reconstruction performance. As in our method, determining the scanning paths online is more effective for the MVS reconstruction.



(a) (b)

(c) (d)

Fig. 10. Reconstruction results of (*top row*) offline and (*bottom row*) online MVS methods. Each model was constructed from the same image set taken by the trajectory of $\text{Ours}_{\text{G+L+O}}$ in scenario 4. The 3-D models show (a), (c) the reconstructed point clouds with (a), (c) original colors and (b), (d) per-point errors coded by color as described in [69]. The offline MVS generated more artifacts than the online MVS around the complex structures (zoomed-in view).



(a) (b)

Fig. 11. Comparative modeling results of the proposed online MVS method and an existing offline MVS method (COLMAP). Each method reconstructed 3-D models by processing two set of images taken by the trajectories of $\text{Ours}_{\text{G+L+O}}$ and Sub-Cov in scenario 4. The online MVS reconstruction without the use of the view selection (w/o VS) was also performed. The graphs show (a) precision and (b) recall curves of each reconstructed model over distance thresholds.

C. Evaluation of Online 3-D Modeling System

To verify the performance of the proposed online MVS method, we conducted a comparative experiment with an offline MVS method in the simulation environment. We used the popular state-of-the-art method, *COLMAP* [3], for offline MVS reconstruction. Both online and offline MVS reconstructions were performed based on two sets of images taken by the trajectories of $\text{Ours}_{\text{G+L+O}}$ and Sub-Cov in scenario 4 in Section VI-A. Furthermore, we separately evaluated the performances of the online MVS method with and without the use of view selection to measure the benefits of the view selection method described in Section V-E. As in Sub-Cov, the online MVS method without view selection determines the source images by extracting the neighboring frames that share the most SLAM map points.

Fig. 11 shows the precision and recall curves of each reconstructed model over distance thresholds τ. The online MVS method generally had higher precision performance than the offline MVS method. The online method applies various outlier filtering methods, including photometric and geometric consistency checks and surfel-weight-based filtering. Therefore, many outliers are removed, so it can obtain high-precision results. Fig. 10 shows the reconstruction results of online and offline MVS with per-point errors coded by color. As can be seen in Fig. 10, the result of online MVS contains significantly fewer artifacts than offline MVS around the complex structures. The outlier filtering approaches in the online MVS could clearly remove outliers even in complex areas.

On the other hand, the online method generally had a lower recall performance than the offline method. The online method selects the source images from insufficient candidate images restricted to previously acquired frames, while the offline method determines the source images from the entire image set. The offline method can use various source images in which many image areas overlap with a reference image. Therefore, the

TABLE III
COMPUTATION TIME OF EACH SUBMODULE AND GPU MEMORY CONSUMPTIONS OF DEPTH ESTIMATION AND SURFEL MAPPING IN THE SIMULATION SCENARIOS

3D Modeling Module			**Path Planning Module**	
Submodules	Time[s]	Mem[MB]	Submodules	Time[s]
Depth Estimation	0.202	2327	Global Planning	1.032
Depth Filtering	0.111		Target Extraction	2.067
Surfel Mapping	0.025	2392	Local Planning	0.853
Loop Closing	9.754		Trajectory Opt.	0.706
Total Comp.	0.338 (+9.754)	4719	Total Comp.	4.658

offline method succeeds in stereo matching for many image areas and can obtain higher recall results than the online method.

The online MVS method with view selection had better performance in both precision and recall than when view selection was not performed. This suggests that view selection considering visibility and geometric factors for a specific target surface, rather than focusing on the entire image area, is effective in complex structure modeling, as in scenario 4.

D. Computation Time Analysis

This section analyzes the computational performances of the proposed method. Table III tabulates the computation times and GPU memory consumptions. Each computation time represents the average time processing an image frame for 3-D modeling or a single planning iteration in the simulation experiments. All submodules were processed on a standard desktop PC with an Intel Core i7-6700 K 4 GHz CPU, 64 GB of RAM, and Nvidia GTX 1080Ti GPU with 11 GB of memory. The 3-D modeling module takes about 0.338 s to process a single image frame and 10.092 s if loop-closing occurs. The offline MVS method [3] takes about 6 h to process 300 images, but our method only takes about 2 min to process the same number of images. Our method also requires 2327 MB of GPU memory to estimate the depth of a 720×480 resolution image and 2392 MB to maintain the maximum 9437 K surfels.

Our path-planning module takes 4.658 ss on average to compute a scanning path at a single iteration. Similar to our method, the NBV method iteratively computes a scanning path by performing the Poisson reconstruction, which takes 2.812 s on average. The NBV only determines a single viewpoint in an iteration, while our method determines multiple viewpoints and trajectory optimization. Compare with the NBV, our method takes more computation time of 1.846 s, but it is not much time considering that our method had much higher modeling performances.

E. Real-World Experiments

We conducted real-world experiments to demonstrate the feasibility of the proposed method in real-world environments. A DJI Matrice-100 drone was used as the MAV platform in the experiments. As shown in Fig. 12(a), we mounted a monocular

Fig. 12. (a) MAV platform with gimbal camera used in the real-world experiments. Experimental environments of (b) Auditorium and (c) Office Building.

camera, a Zenmuse Z3, and an embedded board, a Jetson TX2, on the MAV. The camera captured image frames at a resolution of 1600×900, and the embedded board transmitted the image frames to a laptop at the ground station through LTE data streaming. The laptop processed the online 3D modeling with camera-pose estimation and computed scanning paths. A monocular version of ORB-SLAM was used for pose estimation.

In reality, it is difficult for the MAV to follow a planned path precisely because of several factors, including internal sensor noise and wind gusts. For safety reasons, instead of flying autonomously, the MAV was manually controlled to follow the planned path. An operator checked the planned path and the MAV's moving trajectory online on display and manually controlled the MAV using a controller to follow the trajectory. To achieve accurate pose estimation and precise manual control, we restricted the maximum translational speed to 0.2 m/s and the rotational speed to 0.3 rad/s. We also used a gimbal camera stabilizer for stable image acquisition. To verify if the manually operated trajectories accord the planned trajectories, we performed the trajectory alignment between the two trajectories and calculated the absolute trajectory error [72]. We conducted the experiments several times for each method and then selected the result with the lowest trajectory error for performance evaluation.

We considered two real-world scenarios: modeling a single structure [*Auditorium*; Fig. 12(b)] and modeling multiple structures [*Office Building*; Fig. 12(c)]. The performance of our method was compared with that of Sub-Cov [8]. In the same way as the original explore-then-exploit methods, both the coarse and detailed models for Sub-Cov were computed by the offline MVS program, COLMAP [3]. The total travel budgets of Sub-Cov was set as the path distance result from our method. The explore-then-exploit method has already been widely used in real-world applications; therefore, its modeling results can be an appropriate baseline for the feasibility demonstration.

Figs. 1 and 13 show the reconstruction results of the Auditorium and Office Building scenes, respectively. As can be seen in Fig. 1, the reconstructed models using our method and Sub-Cov are similar and of high quality. They produced a complete model of a single structure (Auditorium) with satisfactory reconstruction. In multiple-structure modeling (Office Building), our method generally had better qualitative performance than Sub-Cov. Our method obtains dense point clouds that represent the entire surfaces of the multiple structures. Even if Sub-Cov performed a thorough scanning to cover all surfaces of the target structures, it is not guaranteed to reconstruct a perfect model due to the inherent weakness of MVS from occlusions and surface textures. Our method, on the other hand, has good reconstruction performances even on complex structures because

Fig. 13. Reconstruction results of the real-world scene (Office Building) obtained using (a) circular trajectory, (b) Sub-Cov, and (c) our method. The top figures show the entire scenes of the reconstructed 3-D models. The bottom figures show close-up renderings of each reconstruction.

it continuously analyzes the reconstruction quality and plans a path for rescanning incompletely reconstructed surfaces. This result demonstrates the feasibility of the proposed 3-D modeling system and path-planning method in real-world environments.

VII. Limitations and Discussion

Although the proposed approach can achieve compelling results in experiments, there are still several major limitations. First, the output of the online 3-D modeling system is very sensitive to localization errors. When scanning a specific region several times with different reference views, a high localization error may produce an inconsistent reconstruction with multilayer surfaces of the region. For accurate pose estimation, we set up texture-rich scenes in the simulation scenarios and used a gimbal camera stabilizer to obtain stable images in real-world experiments. Furthermore, we restricted the maximum limit of the MAV's motion speed to be small. As a result, it was possible to obtain accurate localization results with performance similar to that of offline SfM [53]. However, as shown in Fig. 13, a few minor regions were reconstructed somewhat inaccurately because the SLAM was not possible to estimate perfectly accurate camera poses. Our method generated some incorrectly aligned point clouds and failed to reconstruct several thin structures.

There are several solutions to address this localization problem. Dynamic movements of MAVs can decrease SLAM performance due to tracking errors. This can be mitigated by applying visual-inertial SLAM [73] or event-based SLAM [74]. They aim to measure stable pose information against dynamic and fast motions of MAVs by using IMU or event camera. The dense bundle adjustment method [75] may also be applicable for more precise loop-closing. The method optimizes camera poses and depth estimations simultaneously by solving a dense SfM problem. It enables to obtain a densely registered surface model while preserving the consistency with the pose graph. Another approach can be to consider the localization uncertainty of SLAM for path planning. For future work, active SLAM approaches [76], [77] could be applied to our method to minimize localization errors.

Second, because of memory requirements to perform depth estimation and surfel mapping in our method, the range and resolution of a target 3-D model are limited. This can limit the proposed model to be used onboard. The online MVS [13] performs cost-volume regularization and depth regression using 3-D CNNs, which requires large GPU memory. Therefore, we reduced the image resolution to 1600×900 in the real-world experiments to efficiently process the online MVS on a Nvidia GTX 1080Ti GPU (total 11 GB memory). Furthermore, the surfel mapping method processes a large amount of point cloud data using GPU. The number of surfel points can increase significantly as the image resolution and scene size increase; therefore, the maximum scene size must be limited to the available GPU memory. This can be addressed by dividing the target environment by parts and process independently. First, we need to decompose the entire environment into subregions that can be sufficiently covered given hardware specs. After scanning each subregion independently, we could obtain the final 3-D model of the larger scale target structure by integrating the constructed submodels. Based on this, the proposed method also could be sufficiently processed on an onboard platform. To increase onboard usability, it would be a future study to incorporate a model compression method [78] into the aforementioned method.

Finally, our method only considers the geometric factors for trajectory optimization and view selection. The quality of MVS reconstruction can be affected by various factors, such as texture, dynamic lighting, and shadows; therefore, it is difficult to model the reconstruction quality using only geometric factors. Photometric factors such as visual saliency [79] and image gradient [49] can be additionally considered for view path planning. In addition, 3-D reconstruction uncertainty can also be measured and incorporated with the path planning. Several studies [80], [81] train deep neural networks to obtain reconstruction uncertainty of a scene. The uncertainty enables us to evaluate viewpoints that are likely to provide the most accurate evaluated surface prior to performing MVS. A future work would be to apply the reconstruction uncertainty factor to our path-planning module.

VIII. Conclusion

In this article, we presented a novel framework for autonomous 3-D modeling based on online MVS reconstruction.

The proposed framework incrementally constructs a dense 3-D model of a large-scale structure using an online MVS and surfel-based mapping. The framework also iteratively plans view paths using online feedback based on reconstruction quality. It explores the entire unknown area efficiently while providing comprehensive scanning of local low-quality surfaces. This approach improves the completeness of the reconstructed 3-D models. Furthermore, our method performs trajectory optimization and active image selection based on MVS heuristics to enhance the quality of MVS reconstruction. Our simulation results show that our method has better modeling performance than the explore-then-exploit methods, even with a single exploration trial without rescanning. In particular, our method successfully constructed complete 3-D models of very complex structures. The results of real-world experiments demonstrate the practical feasibility of our method in real-world environments. To the best of our knowledge, this is the first work that implements exploration planning for a MAV to construct high-quality 3-D models based on an online MVS reconstruction.

REFERENCES

[1] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 873–881.

[2] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[3] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis*, 2016, pp. 501–518.

[4] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2006, pp. 519–528.

[5] T. Schops *et al.*, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3260–3269.

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

[7] "3dr Robotics. 3dr Site Scan," 2017. [Online]. Available: https://3dr.com/

[8] M. Roberts *et al.*, "Submodular trajectory optimization for aerial 3D scanning," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5324–5333.

[9] B. Hepp, M. Nießner, and O. Hilliges, "Plan3D: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Trans. Graph.*, vol. 38, no. 1, 2018, Art. no. 4.

[10] C. Peng and V. Isler, "Adaptive view planning for aerial 3D reconstruction," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 2981–2987.

[11] N. Smith, N. Moehrle, M. Goesele, and W. Heidrich, "Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark," in *Proc. SIGGRAPH Asia Tech. Papers*, 2018, Art no. 183.

[12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[13] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2495–2504.

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

[15] S. Song, D. Kim, and S. Jo, "Active 3D modeling via online multi-view stereo," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 5284–5291.

[16] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 2609–2616.

[17] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 333–356, 1988.

[18] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Comput. Surv.*, vol. 35, no. 1, pp. 64–96, 2003.

[19] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *Int. J. Robot. Res.*, vol. 30, no. 11, pp. 1343–1377, 2011.

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

[21] Y. Kwon, D. Kim, I. An, and S.-E. Yoon, "Super rays and culling region for real-time updates on grid-based occupancy maps," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 482–497, Apr. 2019.

[22] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, 1997, pp. 146–151.

[23] S. Shen, N. Michael, and V. Kumar, "Stochastic differential equation-based exploration algorithm for autonomous indoor 3D exploration with a micro-aerial vehicle," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1431–1444, 2012.

[24] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2135–2142.

[25] S. J. Moorehead, *Autonomous Surface Exploration for Mobile Robots*. New York, NY, USA: Carnegie Mellon Univ., 2001.

[26] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Mutual information-based exploration on continuous occupancy maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 6086–6092.

[27] B. Hepp, D. Dey, S. N. Sinha, A. Kapoor, N. Joshi, and O. Hilliges, "Learn-to-score: Efficient 3D scene exploration by predicting view utility," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 437–452.

[28] D. Zhu, T. Li, D. Ho, C. Wang, and M. Q.-H. Meng, "Deep reinforcement learning supervised autonomous exploration in office environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 7548–7555.

[29] B. Charrow *et al.*, "Information-theoretic planning with trajectory optimization for dense 3D mapping," in *Robotics: Sci. Syst.*, vol. 6, pp. 3–12, 2015.

[30] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3D exploration and surface inspection," *Auton. Robots*, vol. 42, no. 2, pp. 291–306, 2018.

[31] C. Wang, H. Ma, W. Chenc, L. Liu, and M. Q.-H. Meng, "Efficient autonomous exploration with incrementally built topological map in 3D environments," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9853–9865, Dec. 2020.

[32] S. Song and S. Jo, "Online inspection path planning for autonomous 3D modeling using a micro-aerial vehicle," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 6217–6224.

[33] S. Song, D. Kim, and S. Jo, "Online coverage and inspection planning for 3D modeling," *Auton. Robots*, vol. 44, pp. 1431–1450, 2020.

[34] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-d slam with volumetric fusion," *Int. J. Robot. Res.*, vol. 34, no. 4/5, pp. 598–626, 2015.

[35] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View/state planning for three-dimensional object reconstruction under uncertainty," *Auton. Robots*, vol. 41, no. 1, pp. 89–109, 2017.

[36] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3D object reconstruction," *Auton. Robots*, vol. 42, no. 2, pp. 197–208, 2018.

[37] R. Monica, J. Aleotti, and S. Caselli, "A kinfu based approach for robot spatial attention and view planning," *Robot. Auton. Syst.*, vol. 75, pp. 627–640, 2016.

[38] S. Chen and Y. Li, "Vision sensor planning for 3-D model acquisition," *IEEE Trans. Syst., Man, Cybern.*, *Part B*, vol. 35, no. 5, pp. 894–904, Oct. 2005.

[39] S. Kriegel, T. Bodenmüller, M. Suppa, and G. Hirzinger, "A surface-based next-best-view approach for automated 3D model completion of unknown objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 4869–4874.

[40] S. Wu *et al.*, "Quality-driven Poisson-guided autoscanning," *ACM Trans. Graph.*, vol. 33, no. 6, 2014, Art. no. 203.

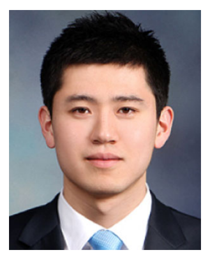
[41] I. D. Lee, J. H. Seo, Y. M. Kim, J. Choi, S. Han, and B. Yoo, "Automatic pose generation for robotic 3-D scanning of mechanical parts," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1219–1238, Aug. 2020.

[42] S. Kriegel, C. Rink, T. Bodenmüller, A. Narr, M. Suppa, and G. Hirzinger, "Next-best-scan planning for autonomous 3D modeling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 2850–2856.

[43] S. Song and S. Jo, "Surface-based exploration for autonomous 3D modeling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1–8.

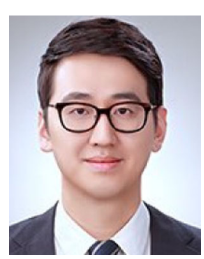
[44] R. Huang, D. Zou, R. Vaughan, and P. Tan, "Active image-based modeling with a toy drone," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1–8.

[45] C. Hoppe *et al.*, "Online feedback for structure-from-motion image acquisition," in *Proc. Brit. Mach. Vis. Conf.*, vol. 2, 2012, pp. 70.1–70.12.
[46] M. Mauro, H. Riemenschneider, A. Signoroni, R. Leonardi, L. J. Van Gool, and I. Brescia, "A unified framework for content-aware view selection and planning through view importance," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.
[47] O. Mendez Maldonado, S. Hadfield, N. Pugeault, and R. Bowden, "Next-best stereo: Extending next best view optimisation for collaborative sensors," *Proc. Brit. Mach. Vis. Conf.*, pp. 65.1–65.12, 2016.
[48] O. Mendez, S. Hadfield, N. Pugeault, and R. Bowden, "Taking the scenic route to 3D: Optimising reconstruction from moving cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4677–4685.
[49] C. Forster, M. Pizzoli, and D. Scaramuzza, "Appearance-based active, monocular, dense reconstruction for micro aerial vehicle," in *Proc. Robot., Sci. Syst. Conf.*, 2014, pp. 1–8, Art. no. EPFL-CONF-203672.
[50] A. Bircher *et al.*, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Auton. Robots*, vol. 40, no. 6, pp. 1059–1078, 2016.
[51] B. Englot and F. S. Hover, "Three-dimensional coverage planning for an underwater inspection robot," *Int. J. Robot. Res.*, vol. 32, no. 9-10, pp. 1048–1073, 2013.
[52] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *Int. J. Comput. Vis.*, vol. 80, no. 2, pp. 189–210, 2008.
[53] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.
[54] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1385–1392.
[55] K. Wang and S. Shen, "Adaptive baseline monocular dense mapping with inter-frame depth propagation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3225–3232.
[56] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 767–783.
[57] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
[58] K. Wang, F. Gao, and S. Shen, "Real-time scalable dense surfel mapping," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 6919–6925.
[59] T. Danner and L. E. Kavraki, "Randomized planning for short inspection paths," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, 2000, pp. 971–976.
[60] B. Hu and G. Raidl, "Effective neighborhood structures for the generalized traveling salesman problem," in *Proc. Evol. Comput. Combinatorial Optim.*, 2008, pp. 36–47.
[61] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, 2000, pp. 995–1001.
[62] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29.1–29.13, 2013.
[63] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
[64] A. Romanoni and M. Matteucci, "Mesh-based camera pairs selection and occlusion-aware masking for mesh refinement," *Pattern Recognit. Lett.*, vol. 125, pp. 364–372, 2019.
[65] M. Popović *et al.*, "An informative path planning framework for UAV-based terrain monitoring," *Auton. Robots*, vol. 44, 6, pp. 889–911, 2020.
[66] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards A New Evolutionary Computation*. Berlin, Germany: Springer, 2006, pp. 75–102.
[67] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. Berlin, Germany: Springer, 2016, pp. 649–666.
[68] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors-a modular gazebo MAV simulator framework," in *Proc. Robot Oper. Syst.*, 2016, pp. 595–625.
[69] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 78.
[70] B. Englot and F. Hover, "Sampling-based coverage path planning for inspection of complex structures," in *Proc. 22nd Int. Conf. Autom. Plan. Scheduling*, vol. 22, no. 1, 2012, pp. 29–37.
[71] B. Englot and F. Hover, "Planning complex inspection tasks using redundant roadmaps," in *Robotics Research*. Berlin, Germany: Springer, 2017, pp. 327–343.
[72] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.
[73] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
[74] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, 2017.
[75] C. Tang and P. Tan, "BA-Net: Dense bundle adjustment network," in *Proc. Int. Conf. Learn. Representations*, 2019.
[76] A. Kim and R. M. Eustice, "Active visual slam for robotic area coverage: Theory and experiment," *Int. J. Robot. Res.*, vol. 34, no. 4-5, pp. 457–475, 2015.
[77] Z. Zhang and D. Scaramuzza, "Beyond point clouds: Fisher information field for active visual localization," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 5986–5992.
[78] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. Int. Conf. Learning Representations*, 2016, *arXiv:1510.00149*.
[79] E. Dunn and J.-M. Frahm, "Next best view planning for active model improvement," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 1–11.
[80] A. Poms, C. Wu, S.-I. Yu, and Y. Sheikh, "Learning patch reconstructability for accelerating multi-view stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3041–3050.
[81] C. Mostegel, M. Rumpler, F. Fraundorfer, and H. Bischof, "UAV-based autonomous image acquisition with multi-view stereo quality assurance by confidence prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 1–10.



Soohwan Song received the B.S. degree in information and communication engineering from Dongguk University, Seoul, South Korea, in 2013, and the M.S. and Ph.D. degrees in computer science from KAIST, Daejeon, South Korea, in 2015 and 2020, respectively.

After graduation, he worked as a Postdoctoral Researcher with Neuro-Machine Augmented Intelligence Laboratory, KAIST. Since March in 2021, he has been with the Intelligent Robotics Research Division, ETRI, as a Senior Researcher. His research interests include robotics, motion planning, and computer vision.



Daekyum Kim received the B.S. degree in mechanical engineering from the University of California, Los Angeles, Los Angeles, CA, USA, in 2015. He is currently working toward the Ph.D. degree in computer science at the Korea Advanced Institute of Science and Technology, Daejeon, South Korea.

His research interests are in the areas of machine learning, computer vision, and robotics.

Sunghee Choi received the B.S. degree in computer engineering from Seoul National University, Seoul, South Korea, in 1995, and the M.S. and Ph.D. degrees in computer science from the University of Texas at Austin, Austin, TX, USA, in 1997 and 2003, respectively.

She has been working as a Professor with the School of Computing, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, since 2004. Her research interests include computational geometry, computer graphics and geometric problems.