2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Surface-based Exploration for Autonomous 3D Modeling

Soohwan Song and Sungho Jo

Abstract— **In this study, we addressed a path planning problem of a mobile robot to construct highly accurate 3D models of an unknown environment. Most studies have focused on exploration approaches, which find the most informative viewpoint or trajectories by analyzing a volumetric map. However, the completion of a volumetric map does not necessarily describe the completion of a 3D model. A highly complicated structure sometimes cannot be represented as a volumetric model. We propose a novel exploration algorithm that considers not only a volumetric map but also reconstructed surfaces. Unlike previous approaches, we evaluate the model completeness according to the quality of the reconstructed surfaces and extract low-confidence surfaces. The surface information is used to guide the computation of the exploration path. Experimental results showed that the proposed algorithm performed better than other state-of-the-art exploration methods and especially improved the completeness and confidence of the 3D models.**

I. INTRODUCTION

Precise 3D models of large structures are increasingly required in many industrial applications, such as the digitization of cultural heritage, renovation design, and structural inspection. Traditionally, the modeling process has been performed manually; however, some studies in recent years have tried to use mobile robots to automatically generate the models [1] [2] [3]. For the autonomous task, mobile robots iteratively compute the optimal views to reconstruct a structure and complete the model by sensing the views. This is known as a next-best-view (NBV) problem [4]. If the mobile robots model an unknown environment, this is defined as an exploration problem.

In this study, we investigated a path planning problem for mobile robots to construct highly accurate 3D models of unknown structures. Most studies [1] [2] [3] [5] have used a volumetric reconstruction method [6] to represent the environment as a volumetric map. A mobile robot explores the unknown environment while simultaneously constructing the volumetric map until it reaches a certain percentage of completion. The volumetric map has some advantages of facilitating a visibility check of certain volumes from the ray-cast operation and allowing the probabilistic occupancy estimation. Furthermore, free space is directly accessed from the volumetric map; this is particularly efficient for planning a collision-free path. After the exploration is completed, the point cloud streams recorded from a sensor are post-processed to obtain a complete 3D surface model [2]. However, even if the volumetric map is complete, it does not necessarily mean that the quality of the surface model is also perfect. It is difficult to represent the surfaces of a very complex model as only free or occupied volumes. Therefore, the volumetric map is inefficient for completing 3D models and may provide poor-quality reconstructed surfaces.

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

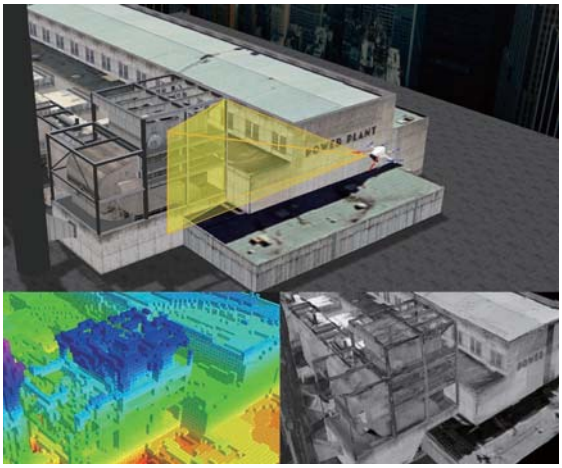


Fig. 1. Example of autonomous 3D modeling of Power Plant structure. The robot explores around the structure while the structure is scanned with the attached vision sensor (*upper*). Our approach computes the exploration path by analyzing the volumetric (*lower left*) and surface model (*lower right*) at the same time.

To address this problem, we propose a novel exploration algorithm that considers not only the volumetric map but also reconstructed surfaces. Unlike previous approaches, we evaluate the model completeness by analyzing the quality of the reconstructed surface. The algorithm first extracts inaccurately reconstructed surfaces by analyzing the quality and trend of the surfaces from the truncated signed distance fields (TSDFs) [7]. Frontiers are also extracted from the volumetric map. Next, the algorithm plans an inspection path that fully covers both the low-quality surfaces and frontiers. The inspection path is iteratively refined according to the updated surfaces and volumes until a specific local area is completely modeled. Our algorithm then fully analyzes the 3D graphical surfaces and efficiently explores the unknown region at the same time.

The remainder of this paper is structured as follows. Section II presents the related works on mobile robot exploration and autonomous modeling. Section III defines the considered problem, and Section IV describes the proposed approach in detail. Section V gives the experimental results, and Section VI provides the conclusions and future works.

II. RELATED WORKS

NBV algorithms automatically compute a new viewpoint to take the most informative sensor measurements from the

978-1-5386-3081-5/18/$31.00 ©2018 IEEE

environment. Recently, they have been extensively employed in the field of robotics due to the growth of practical applications, including autonomous 3D modeling and environment exploration tasks. The estimation from an NBV algorithm can be used to guide a robot to plan a sensing strategy in various autonomous systems. According to Scott et al. [8], NBV algorithms can be classified into two categories: volumetric and surface-based methods.

Volumetric methods determine an NBV by analyzing spatial information from a volumetric map. As noted earlier, the volumetric map is appropriate for exploration tasks because of its accessibility of spatial information that can be used for collision avoidance, ray casting, and so on. Yamauchi [9] first defined frontiers as a boundary between the explored and unexplored areas in a map. The frontier concept has been applied to volumetric maps and used in various 3D exploration and modeling algorithms [2] [3] [5] [10] [11] [12]. Shen et al. [10] used a frontier-based method with a particle-based representation of 3D space. The expansion of a particle system is related to the unexplored region, and it enables a mobile robot to explore an indoor environment. Cieslewski et al. [11] extended the original frontier-based method [9] to explore as much as possible at high speed. The method determines a goal frontier from the current field of view, reducing changes in velocity. Bircher et al. [2] employed a receding horizon planning strategy for the exploration algorithm, which samples feasible trajectories using a rapidly exploring random tree (RRT) and moves to the first node of the best trajectory. In our previous work [12], we proposed an online inspection method for the exploration and modeling of unknown environments. The algorithm efficiently explores local unknown regions and improves the completeness of constructed volumetric models. In this work, we extend the online inspection method [12] to additionally account for the surface information of a model.

Surface-based methods evaluate viewpoints by analyzing the shape of the surfaces constructed so far. Such methods focus on completing the surface model, which consists of a surface mesh or high-resolution point cloud. Most surface-based methods [13] [14] [15] are generally utilized to model small-scale objects, not large structures. Chen and Li [14] estimated the surface trend of a target object in an unknown region. Their method can be applied to simple and smooth surfaces, but it is not applicable to complex structures. Wu et al. [15] measured the confidence of surfaces estimated from Poisson surface reconstruction [16]. The confidence is defined as the completeness and smoothness of the constructed iso-surfaces. Their approach uses the confidence values to generate an ambient viewing field, which provides a set of NBVs for covering low confidence regions. Kriegel et al. [17] proposed a combined approach of surface-based and volumetric methods. They first generated a set of scan paths from the surface trends estimated by boundary detection and then evaluated the information gain with the volumetric model.

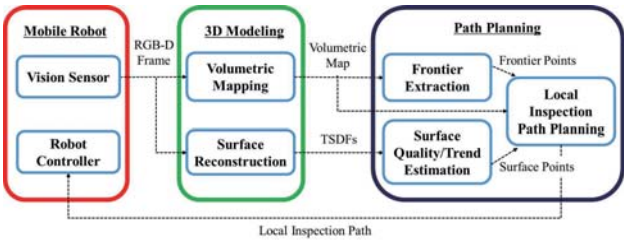


Fig. 2. Overall system architecture of the proposed framework.

A. Contributions

State-of-the-art approaches to autonomous modeling of unknown large environments [2] [3] [5] [11] [12] have focused on exploring the volumetric map while disregarding the reconstructed surface quality. On the other hand, surface-based methods [13] [14] [15] can be used to analyze the reconstruction quality but have been mostly applied to precise modeling applications of small objects. We propose a method to generate a complete surface model by applying a surface-based method to an exploration algorithm.

The contributions of this paper can be summarized as follows: (i) Unlike past solutions, we apply a surface-based method to the exploration problem in order to reconstruct a complete surface model. (ii) We provide an efficient inspection algorithm that provides full coverage of surfaces and volumes by applying different sampling methods to each model. (iii) We propose a method to compute the sector visitation order. This method provides a global coverage sequence of a 3D unknown environment and can minimize the number of sector revisits. (iv) We empirically evaluated our approach in two simulated scenarios: modeling a single structure and modeling multiple structures. The results proved the effectiveness and applicability of our method.

III. PROBLEM DESCRIPTION

In this study, we considered the problem of a mobile robot exploring an unknown and spatially bounded 3D space $V \subset \mathbb{R}^3$ while ensuring high-quality surface reconstruction of structures in the space. We assumed that the robot is equipped with a forward-looking vision sensor such as a stereo camera or Kinect sensor and RGB-D data are collected from the sensor. The sensor has some constraints, such as the max/min sensing ranges, field of view, and minimum incidence angle. The estimated RGB-D data are integrated into a probabilistic volumetric map $\mathcal{M}$ and used to construct a surface model $\mathcal{F}$ at the same time. The volumetric map is constructed with the Octomap framework [6] and represents the workspace in three states: occupied ($V_{occupied} \subset V$), free ($V_{free} \subset V$), and unknown ($V_{unknown} \subset V$). The surface model is constructed from TSDFs, which implicitly represent surfaces as high-resolution fields. The TSDFs can filter sensor noise from a large amount of sensing data and quickly estimate the surface point cloud with the zero-crossing method [7]. The ultimate objective is to fully explore the unknown space in the volumetric map and construct a high-quality surface model in a short period of time.

Algorithm 1 Proposed path planning algorithm

Input: Volumetric map $\mathcal{M}$, TSDFs $\mathcal{F}$, Current configuration q_{curr}, Current sector s_{surr}, and Sector set S.

1: $s_{next} \leftarrow GetNextSector(S, s_{curr})$
2: $R_{search} \leftarrow GetSearchRegion(s_{curr}, s_{next})$
3: $q_{goal} \leftarrow ComputeGoalConf(q_{curr}, s_{next}, R_{search})$
4: **while** $q_{curr} \neq q_{goal}$ **do**
5: $[R^{surf}_{search}, R^{non-surf}_{search}] \leftarrow DivideParts(\mathcal{M}, R_{search})$
6: $P_{surf} \leftarrow GetSurfacePoints(\mathcal{F}, R^{surf}_{search})$
7: $X_{front} \leftarrow GetFrontierPoints(\mathcal{M}, R^{non-surf}_{search})$
8: **if** $\#(V^{new}_{front}) > \theta_{front}$ **then**
9: $\xi_{local} \leftarrow InspectionPathPlanning(q_{curr}, q_{goal}, P_{surf}, X_{front}, R_{search})$
10: **end if**
11: $MoveToward(\xi_{local})$
12: $Update(\mathcal{M}, \mathcal{F}, q_{curr})$
13: **end while**

We denote the configuration of the mobile robot as $q \in Q$, where $Q \subseteq R^n$ is the configuration space. We assumed that the localization of the robot is precise, so the exact configuration can be accessed. The path $\xi : [0, 1] \rightarrow Q$ is defined as a sequence of configurations. It is computed only within the known free space V_{free} in $\mathcal{M}$, which guarantees collision-free navigation.

IV. PROPOSED APPROACH

To consider the problem of surface reconstruction and exploration of an unknown environment, the proposed approach plans an exploration path by analyzing the volumetric map $\mathcal{M}$ and surface model $\mathcal{F}$. We decompose the whole space into equal-sized cuboid regions denoted as sectors S. The mobile robot consistently computes the visitation order of the sectors and sequentially completes the entire model according the visitation order. Therefore, we can reduce the number of revisits of the sectors and the total length of the robot trajectory. For each sector, we divide the volumetric map into two parts: a surface-containing region and a non-surface region. In the surface-containing region, the method focuses on improving the reconstruction quality by analyzing the TSDFs. In the non-surface region, the volumetric approach is employed to account for the exploration of the unknown regions on the volumetric map.

Fig. 2 shows an overview of our framework tested, and Algorithm 1 shows the pseudocode of the proposed algorithm. We adopted an online inspection methodology [12], which first determines a goal configuration and then iteratively plans a local inspection path to cover the local region. The algorithm first determines the next sector s_{next} by computing the visitation order of the sectors in the current map (line 1 and Section IV.A). We define the search region R_{search} as an area inside the sectors starting from s_{curr} to s_{next} (line 2). The algorithm determines the goal configuration q_{goal} that maximizes the information gain and closes to the center of s_{next} at the same time (line 3 and Section IV.B). The robot moves to q_{goal} and simultaneously plans the local inspection path ξ_{local} to completely reconstruct surfaces in R_{search}. The local path is iteratively refined according to the updated surfaces until the modeling in the local area is complete (lines 4-13).

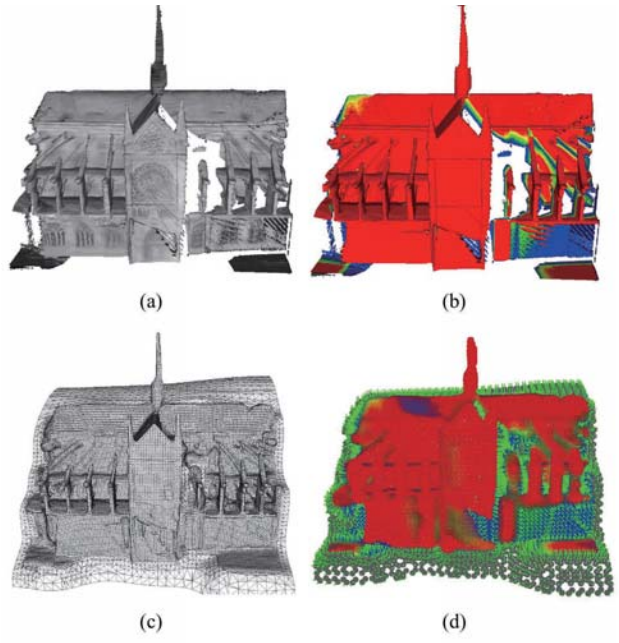


Fig. 3. The extraction process of target surface points: We first obtain an incomplete point cloud (a) and its TSDF weight value (b) by detecting the zero-crossing of TSDFs. The high-weight points are shown in red and low weight in blue. We then reconstruct the tentative surface model (c) using screened Poisson reconstruction algorithm. Finally, the target surface points, green normal points, are estimated by extracting the low confidence surfaces (d). The gray points represent the zero-confidence surfaces.

To compute the local path, our approach first divides the volumetric map in R_{search} into two parts: the surface-containing region R^{surf}_{search} and non-surface region $R^{non-surf}_{search}$ (line 5). R^{surf}_{search} is a region that contains the occluded cells and their surrounding cells at a specific distance. The remaining region excluding R^{surf}_{search} is $R^{non-surf}_{search}$. The quality and trend of the surfaces in R^{surf}_{search} are estimated from $\mathcal{F}$, and low-quality surface points P_{surf} are extracted (line 6 and Section IV.C). We then extract the frontier points X_{front}, which are set at the center positions of the frontiers in $R^{non-surf}_{search}$ on $\mathcal{M}$ (line 7). After P_{surf} and X_{front} are extracted, our algorithm plans an inspection path that fully covers them (line 9 and Section IV.D). The inspection path is iteratively refined according to the updated surfaces and volumes. The refinement step of the inspection path is performed only if the number of new frontier points $\#(X^{new}_{front})$ is greater than the constant value θ_{front} (line 8). If the robot reaches q_{goal}, the iteration stops. The path planning approach is repeated until the whole model is completely constructed.

A. Sector visitation order

We use the information of the current volumetric map to compute the visitation order of all remaining sectors except for the already passed sectors. In this section, we regard

the unknown space in $\mathcal{M}$ as free space because the path between sectors is calculated in a partially known environment (most are unknown). To solve the problem, we use a similar approach to that of Das et al. [18], who transformed the original problem into a Hamiltonian path problem by applying metric closure. The metric closure method is used to construct an augmented graph $G^{aug} = (V^{aug}, E^{aug})$ from an adjacency graph. Each vertex $v_i^{aug} \in V^{aug}$ is located at the center of the free space in an unvisited sector s_i. Each edge $e_{ij}^{aug} \in E^{aug}$ represents the connection of vertices v_i^{aug} and v_j^{aug}, and it has a cost value c_{ij}^{aug}, which is defined as the shortest path cost between the vertices. G^{aug} is a complete graph, which is obtained by finding the all-pair shortest paths in the adjacency graph.

To efficiently compute the shortest path in a 3D space, we construct a sparse adjacency graph $G^{adj} = (V^{adj}, E^{adj})$ from the octree structure [19]. The vertex set V^{adj} is composed of free octree nodes whose resolution is greater than the specific resolution $\theta_{adj-res}$. The edge set E^{adj} is estimated with the neighbor search algorithm from Xu et al. [19]. The number of vertices in G^{adj} is significantly smaller than that of a regular 3D grid. Thus, we can reduce the computation time of the graph search algorithm. Finally, we estimate the all-pair shortest paths of V^{aug} by using an A* search in G^{adj} and then compute the Hamiltonian path in G^{aug} by using a heuristic TSP solver [20].

B. Goal determination

Similar to our previous work [12], we generate a set of feasible samples by extending branches of a rapidly exploring random tree T [21] and select the sample with the most information gain as the goal q_{goal}. The samples are only generated in the search region R_{search}. We define the information gain $Gain(q_k)$ as the volume of unknown cells that can be observed at q_k, which is penalized by two distance factors D_T and D_E:

$$Gain(q_k) = Vis(R_{search}, q_k) e^{-\lambda \cdot \{D_T(q_{curr}, q_k) + D_E(q_k, s_{next})\}} \quad (1)$$

where $Vis(R_{search}, q_k)$ is the volume of visible and unknown cells from q_k in R_{search}. The parameter λ penalizes the long distance factors. $D_T(q_{curr}, q_k)$ is the distance of a path starting from q_{curr} to q_k in the constructed random tree T, and $D_E(q_k, s_{next})$ is the Euclidean distance for the position of q_k and center of the sector s_{next}. These two distance factors give a higher weight to the sample trajectory T towards the center of next sector. Therefore, the determined goal configuration q_{goal} covers a large amount of the unknown volume in R_{search} and closes to the center of s_{next} at the same time.

C. Surface model analysis

In this section, we describe how to extract target surface points that need to be sensed for the high-quality surface model. The TSDFs are composed of equal-sized voxels, and each voxel v contains a signed distance value d_v and weight value w_v. d_v represents the distance to the closest surface, and w_v represents the reliability of the surface measurement. We apply the TSDF integration method, as given by Newcombe et al. [7], to accumulate a simple constant weight 1 up to the maximum weight w_{max}. The estimated surface point cloud can be extracted by detecting the zero-crossings of the TSDFs (Fig. 3a).

After the surface point cloud is generated, we can reconstruct a tentative surface model for the structures by using the screened Poisson surface reconstruction algorithm [16] (Fig. 3c). For real-time computation, we sample the point cloud on a voxel grid with a resolution that is four times as less as that of the TSDFs. The reconstructed surfaces are represented as a set of oriented points $P = \{(x_i, n_i)\}_{i=1,...,N}$, where x_i is the center position of a surface and n_i is a surface normal. We refer to them as surface points. To evaluate the quality of the reconstructed surfaces, we define the confidence of a surface point p_i as the average weight of neighbor points:

$$Conf(p_i) = \frac{\sum_{k \in N(p_i)} \overline{w}_k}{\#(N(p_i))} \quad (2)$$

where $N(p_i)$ is the neighbor points of p_i. The $\overline{w}_k$ is a normalized weight that ranges from zero to one; it is defined as $\overline{w}_k = w_k / w_{max}$. If there is no neighbor point, we set the confidence value to zero. Finally, the surface points whose confidence values are lower than $\theta_{low-conf}$ are determined as the target surface points P_{surf}. The extracted target surface points are shown in Fig. 3d.

D. Inspection path planning

In this section, we address the inspection problem for both the surface points P_{surf} and frontier points X_{front}. The visibility of X_{front} is estimated through ray-casting in the view frustum from a viewpoint. However, the visibility of P_{surf} is additionally considered for the incidence angle. The angle between the surface normal and view direction has to be smaller than the parameter of the minimum incidence angle $\theta_{min-inc}$. Thus, the inspection problem of P_{surf} is more complicated than that of X_{front}; they should be solved in different ways.

We employ the sampling-based approach [22], which finds a set of viewpoints that fully cover the target structures by random sampling. There are two sampling methods for inspection: the primal method [12] [23] [24] randomly samples

Algorithm 2 Local inspection path planning algorithm

Input: Current configuration q_{curr}, Goal configuration q_{goal}, Surface points P_{surf}, Frontier points X_{front}, and Search region R_{search}.

/* Surface inspection planning */
$C_{surf} \leftarrow ClusterSurfacePoints(P_{surf})$
$\{Q_1, ..., Q_N\} \leftarrow DualSampling(C_{surf}, R_{search})$
$\{q_{\pi_1}, ..., q_{\pi_N}\} \leftarrow SolveGTSP(\{Q_1, ..., Q_N\}, q_{curr}, q_{goal})$
$\xi_{surf} \leftarrow GetLinearPath(q_{curr}, \{q_{\pi_1}, ..., q_{\pi_N}\}, q_{goal})$
/* Frontier inspection planning */
$P_{front} \leftarrow GetUnVisibleFront(\{q_{\pi_1}, ..., q_{\pi_N}\}, P_{front})$
$\xi_{local} \leftarrow FrontINSPPath(\xi_{surf}, \{q_{\pi_1}, ..., q_{\pi_N}\}, P_{front})$
return ξ_{local}

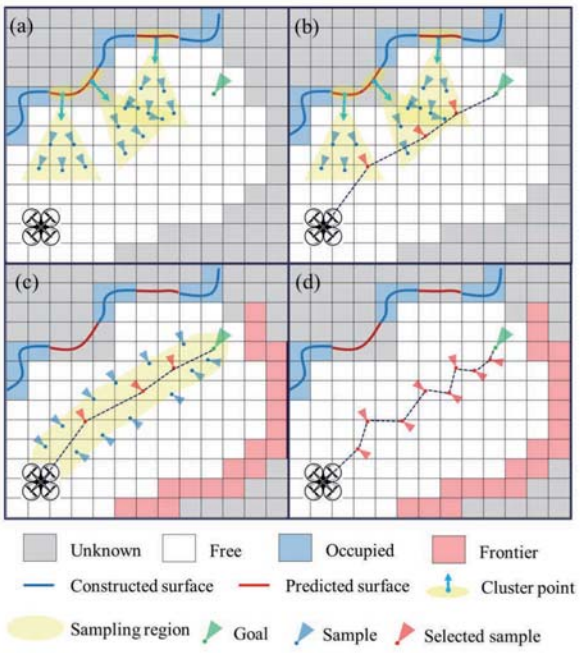


Fig. 4. Overview of the process of local inspection path planning algorithm. The processes are described in 2D for clarity. The algorithm first generates coverage samples of cluster points by the dual sampling (a) and solves the GTSP to determine the optimal tour path (b). We then refine the path to cover the frontiers using primal sampling algorithm [12] (c). The final local inspection path is shown in (d).

the configuration space until all targets are fully covered, and the dual method [25] [26] inversely selects a target and samples the configurations in the cover region where the target is visible. The cover region of a surface point can be reduced by the constraint of the incidence angle, so the dual method is more efficient for P_{surf} than the primal method. On the other hand, X_{front} has a wide-ranging cover region, so it is suitable for the primal method. Therefore, we apply the primal and dual sampling methods to X_{front} and P_{surf}, respectively.

We propose a two-step inspection planning approach to cover both P_{surf} and X_{front}. The first step generates coverage samples of P_{surf} by the dual sampling method and then plans the surface inspection path ξ_{surf} by using the samples. The path starts from a current configuration, covers the entire surface points on P_{surf}, and ends in the goal configuration. The second step refines ξ_{surf} to fully cover the frontier points X_{front} near the path by using the primal sampling method. Algorithm 2 shows the pseudocode of the proposed approach, and Fig. 4 illustrates the path planning process.

1) Surface inspection planning: To efficiently plan an inspection path in real time, we process clustered surface points rather than each individual point. We greedily cluster multiple nearby surface points by randomly selecting a surface point and grouping it with all other points within a distance θ_{dist}^{clust} and angle θ_{angle}^{clust} between their normals. We repeat this clustering process until every surface point belongs to a cluster and reject the clusters with small number

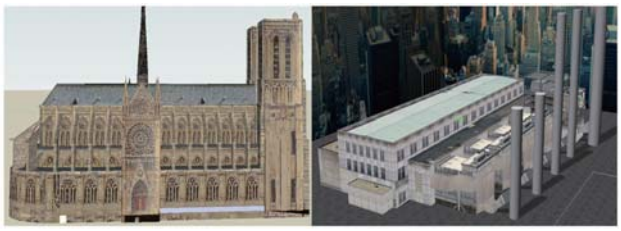


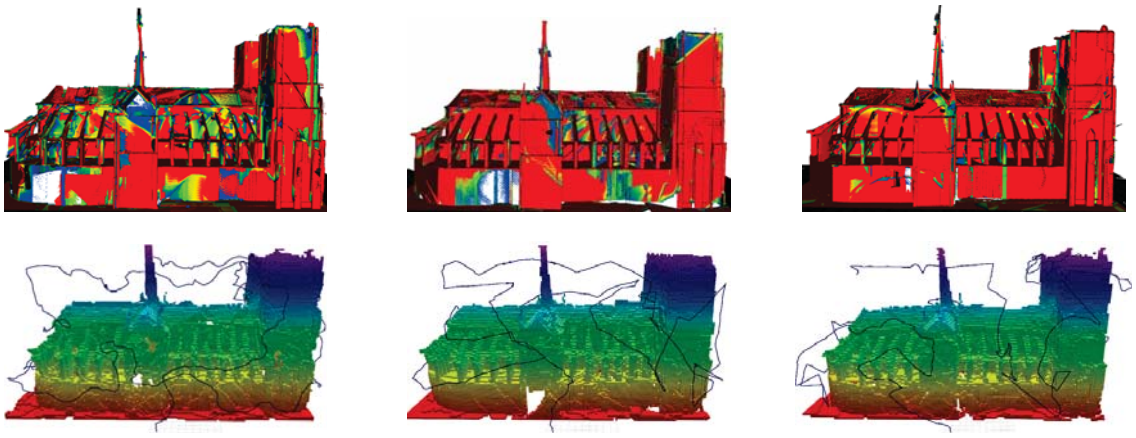
Fig. 5. Two simulated environments in the ROS simulator: (*left*) Notre Dame Cathedral model and (*right*) Power Plant model.

of surface points (line 1). We define a surface cluster $c_i \in C_{surf}$ as the averaged surface point of the clustered points. For each clustered point c_i, we generate a sample set $Q_i \subset Q$ by the dual sampling method [25] (line 2). The dual sampling method inversely estimates a cover region where a certain cluster point c_i is visible and uniformly generates samples in the region. The cover region of c_i is estimated by composing a view frustum from c_i to its normal direction and ray casting into the view frustum for a visibility check. Let d_{dir} be the distance of a direct path from q_{curr} to q_{goal}. A sample q_i is rejected if the total distance of a path from q_{curr} to q_{goal} transferring q_i is greater than $1.5 \times d_{dir}$. Furthermore, the samples outside R_{search} are also rejected. A cluster c_i that does not have any feasible sample is excluded from the cluster set C_{surf}.

After the dual sampling, we extract an inspection path of the surface points by finding the minimum distance trajectory that visits at least one sample $q_i \in Q_i$ from each sample set. This is a generalized traveling salesman problem (GTSP). Given a set of sample sets $\{Q_1, ..., Q_N\}$, the GTSP finds the minimum cost tour starting from q_{curr}, visits exactly one sample q_i per sample set Q_i (line 3), and ends at q_{goal}. We represent the optimal tour as a sequence of selected sample configurations $\{q_{\pi_1}, ..., q_{\pi_N}\}$, where $\Pi = \{\pi_1, ..., \pi_N\}$ is the permutation of the cluster indices $\{1, ..., N\}$ representing the visitation order. This approach is similar to Hess et al.'s approach [26], which transforms the GTSP into a TSP [27] and then computes the solution for the TSP. However, we use the generalized 2-opt neighborhood method [28] for computational efficiency. The distance of each configuration is defined as the Euclidean distance directly connecting the pairs. If the connection has a collision, the A* planner is used to connect them.

2) Frontier inspection planning: The planning algorithm of the local inspection path in [12] is employed to entirely cover the frontier points on X_{front}. The algorithm [12] first computes the shortest path from the current position to the goal and iteratively generates coverage samples within a certain distance d_{sample} from the path. The algorithm utilizes a streaming set cover approach to maintain a suboptimal coverage solution in each iteration. The suboptimal solution is used to decrease the size of the sampling domain to possibly improve the solution.

We modified the original inspection algorithm [12] to sequentially follow the sampled configurations $\{q_1, ..., q_N\}$ along ξ_{surf}. First, the visible frontier points from each con-



(a) RH-NBV (b) Online-INSP (c) SBE

Fig. 6. Results in Scenario 1. Constructed surface models (*upper*) and volumetric models of the Notre Dame Cathedral object with trajectories taken by the MAV (*lower*) at the end of executions of RH-NBV [2] (a), Online-INSP [12] (b), and the proposed SBE (c). Each surface model was color-coded according to the surface confidence; the high-confidence surfaces are shown in red and low confidence in blue.

TABLE I
PARAMETERS USED IN THE EXPERIMENTS.

Parameter	Value	Parameter	Value
Resolution of $\mathcal{F}$	$0.03125m$	θ_{front}	100
Resolution of $\mathcal{M}$	$0.3m$	$\theta_{adj-res}$	$2.4m$
Size of sector	$10{\times}10{\times}5m^3$	$\theta_{low-conf}$	0.4
λ	0.5	$\theta_{min-inc}$	$60°$
d_{sample}	$1.5m$	θ^{clust}_{dist}	$1m$
RRT edge length	$1m$	θ^{clust}_{angle}	$30°$

figuration $q_k \in \{q_{\pi_1}, ..., q_{\pi_N}\}$ are excluded from P_{front} (line 5). Instead of computing the shortest path, we directly utilize the computed path ξ_{surf} and generate the coverage samples near ξ_{surf}. Similar to the original algorithm, we then use the online set cover approach to determine the coverage solution Q^*. The configuration set $\{q_{\pi_1}, ..., q_{\pi_N}\}$ is integrated into Q^*. Finally, the inspection path is computed by extracting the shortest connecting path over all configurations in Q^* using a TSP solver [20]. The algorithm then shortens the path length by using Englot et al.'s heuristic speed-up improvement method [22].

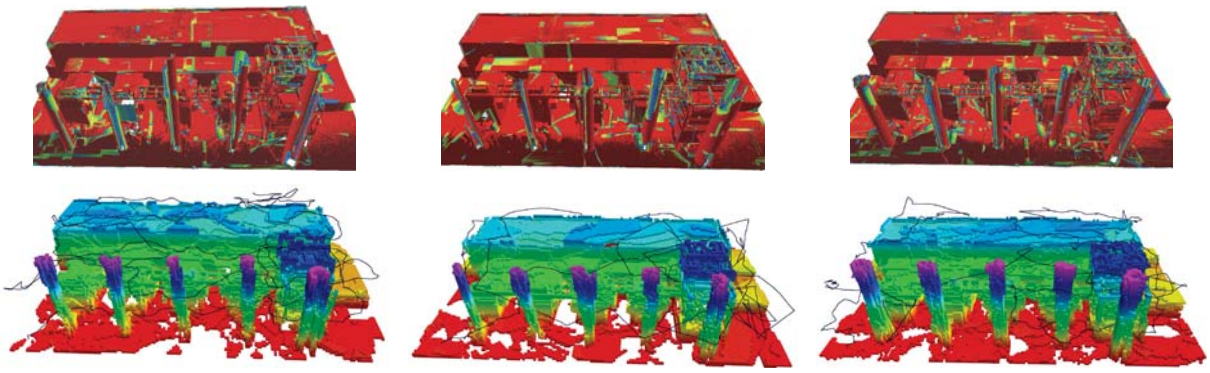
V. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed method, simulation experiments have been performed using the RoterS simulator [29] with the micro-air vehicle (MAV) model of the Firefly hexa-copter. We assume that the MAV is in a flat state with zero roll and pitch [2], which results in a 4D configuration space comprising 3D coordinates and yaw angle. We limit the maximum translational speed to $v_{max} = 0.2m/s$ and rotational speed to $\dot{\psi}_{max} = 0.5rad/s$. These limits are small enough for accurate surface reconstruction and exact path following. A forward-looking stereo camera was mounted on the MAV. It had a max/min sensing range of $[0.3m, 8m]$ and field of view of $[60°, 90°]$ in the vertical and horizontal directions. To integrate the acquired RGB-D data into the TSDFs, we employed the Kintinuous system [30]. The system can construct a dense surface model over an extended region by virtually translating TSDFs. For computational efficiency, the surface models were analyzed only with the TSDF shifting volume in the Kintinuous system. We set the size of the shifting volume to $16 \times 16 \times 16m^3$.

We considered two simulation scenarios: modeling a single structure (*Notre Dame Cathedral*[1]) and modeling multiple structures (*Power Plant*[2]). All structures in the scenarios were relatively large and more complex than the model used in previous studies [2] [12]. Fig. 5 shows each simulation environment. The proposed approach of *surface-based exploration* (SBE) was compared with the *receding horizon NBV* algorithm (RH-NBV) [2] and *online inspection* approach (Online-INSP) [12]. We used the same parameter settings for RH-NBV, Online-INSP, and SBE, such as the edge length of RRT and λ. Table I summarizes all parameters used in every scenario.

For the comparison, we evaluate the performances from two perspectives: exploration performance and modeling quality. To evaluate the exploration performance, we computed the completion time and total path length. When the mobile robot covered 90% of the actual free space, we stopped the exploration and evaluated the performance. We also evaluated the modeling quality from the surface coverage and average confidence of the reconstructed surface points. The surface coverage was computed by matching the reconstructed point cloud to that of the original model [31]. If the distance between an original point and its closest reconstructed point was less than the registration distance of $0.05m$, the original point was considered as observed. The surface coverage is the percentage of the observed point cloud over the entire original point cloud. We defined the confidence of the reconstructed surface point as its TSDF

[1]http://3dwarehouse.sketchup.com/
[2]http://models.gazebosim.org/



(a) RH-NBV (b) Online-INSP (c) SBE

Fig. 7. Results in Scenario 2. Constructed surface models (*upper*) and volumetric models of Power Plant with trajectories taken by the MAV (*lower*) at the end of executions of RH-NBV [2] (a), Online-INSP [12] (b), and the proposed SBE (c).

TABLE II
EXPERIMENTAL RESULTS OF TWO SCENARIOS.

Algorithm	Scenario 1			Scenario 2		
	RH-NBV	Online-INSP	SBE	RH-NBV	Online-INSP	SBE
Completion. Time(min)	62.92	**55.28**	57.88	41.26	**30.56**	34.67
Path Length(m)	583	**506**	521	376	**286**	324
Coverage (%)	87.75	93.36	**97.50**	85.59	90.45	**96.32**
Avg. Confidence	0.77	0.80	**0.84**	0.80	0.85	**0.88**

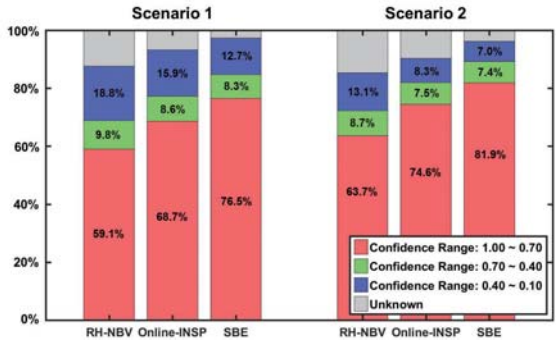


Fig. 8. The confidence distributions of each constructed surface model.

weight normalized to the range [0, 1].

A. Simulation results

We assumed that the bounded space containing the target structures is provided and the remaining space is free. The size of the bounded space was defined as $22 \times 46 \times 25m^3$ for Scenario 1 and $50 \times 32 \times 12m^3$ for Scenario 2. The MAV constructed volumetric maps and surface models of the structure while exploring the bounded spaces. Table II tabulates the results of every scenario as the average of 10 executions. Figs. 6 and 7 show the paths and constructed volumetric and surface models after the execution of each best case for Scenarios 1 and 2, respectively.

Online-INSP showed the best exploration performance in terms of completion time and path length. However, considering that SBE generated complex local paths for the surface coverage, the performance gap between SBE and Online-INSP was not significant. SBE consistently planned a visitation sequence of the sectors. Thus, it improved the exploration performance by reducing the number of revisits of the sectors. RH-NBV and Online-INSP focused only on planning the local path, so their trajectories sometimes overlapped in some regions. As shown in Figs. 6 and 7, RH-NBV generated a complex path that frequently overlapped while the path computed by Online-INSP sometimes overlapped. Even if Online-INSP fully covered the local area, the MAV sometimes moved through regions that it already covered to arrive at the next unexplored region. However, in the case of SBE, the MAV rarely passed through already explored regions, reducing the total exploration time.

SBE had the best modeling performances in terms of the surface coverage and average surface confidence. Although Online-INSP was efficient at volumetric modeling, it did not completely cover the surfaces of a complex structure. Occlusions frequently occur in complex structures. Thus, the surfaces from the complex structures may not be reconstructed because of the incidence angle factor, even if the volumetric map recognizes it as an occupied cell. On the other hand, SBE had 97.50% and 96.32% surface coverages, which indicates that the complex structures can be covered by a surface-based approach. In particular, SBE generated surface models with the highest average confidence; 0.84% and 0.88%. Fig. 8 shows the confidence distributions of each constructed surface model. The high confidence surfaces were distributed over a large percentage, as shown in Figs. 6c and 7c. These results indicate that SBE is suitable for high-quality surface modeling systems.

VI. CONCLUSION AND FUTURE WORKS

We present a novel exploration method to generate a complete surface model in an unknown environment. The

method sequentially visits sectors and completes the entire model. The key is to analyze the surface trend and quality on TSDFs and use the results to guide the computation of the exploration path. When moving to each sector, the robot iteratively plans an inspection path to cover both the low-confidence surfaces and frontiers. As a result, we can efficiently explore an environment and construct a high-quality surface model at the same time. Experimental results showed that our method performed better than other exploration methods and especially improved the completeness and confidence of the constructed surface models.

For future work, we will evaluate the proposed approach in real-world experiments with a real mobile robot. Our approach in this paper assumed that the exact pose of a robot can be estimated and that the robot can follow a planned path accurately. However, this assumption is not always valid in real-world applications; we should consider the uncertainty of the localization and motion of the robot.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education (NRF-2016R1D1A1B01013573).

REFERENCES

[1] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View planning for 3d object reconstruction with a mobile manipulator robot," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 4227–4233.

[2] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon" next-best-view" planner for 3d exploration," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1462–1468.

[3] L. Yoder and S. Scherer, "Autonomous exploration for infrastructure modeling with a micro aerial vehicle," in *Field and service robotics*. Springer, 2016, pp. 427–440.

[4] C. Connolly, "The determination of next best views," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 432–435.

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

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

[7] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.

[8] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys (CSUR)*, vol. 35, no. 1, pp. 64–96, 2003.

[9] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*. IEEE, 1997, pp. 146–151.

[10] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3d exploration with a micro-aerial vehicle," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 9–15.

[11] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight."

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

[13] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1016–1030, 1999.

[14] S. Chen and Y. Li, "Vision sensor planning for 3-d model acquisition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 5, pp. 894–904, 2005.

[15] S. Wu, W. Sun, P. Long, H. Huang, D. Cohen-Or, M. Gong, O. Deussen, and B. Chen, "Quality-driven poisson-guided autoscanning," *ACM Transactions on Graphics*, vol. 33, no. 6, 2014.

[16] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 3, p. 29, 2013.

[17] S. Kriegel, C. Rink, T. Bodenmüller, A. Narr, M. Suppa, and G. Hirzinger, "Next-best-scan planning for autonomous 3d modeling," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2850–2856.

[18] A. Das, M. Diu, N. Mathew, C. Scharfenberger, J. Servos, A. Wong, J. S. Zelek, D. A. Clausi, and S. L. Waslander, "Mapping, planning, and sample detection strategies for autonomous exploration," *Journal of Field Robotics*, vol. 31, no. 1, pp. 75–106, 2014.

[19] S. Xu, D. Honegger, M. Pollefeys, and L. Heng, "Real-time 3d navigation for autonomous vision-guided mavs," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 53–59.

[20] K. Helsgaun, "An effective implementation of the lin–kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.

[21] S. Karaman and E. Frazzoli, "Sampling based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[22] B. J. Englot and F. S. Hover, "Sampling-based coverage path planning for inspection of complex structures," in *Twenty-Second International Conference on Automated Planning and Scheduling*, 2012.

[23] B. Englot and F. Hover, "Planning complex inspection tasks using redundant roadmaps," in *Robotics Research*. Springer, 2017, pp. 327–343.

[24] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Asymptotically optimal inspection planning using systems with differential constraints," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4126–4133.

[25] T. Danner and L. E. Kavraki, "Randomized planning for short inspection paths," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 971–976.

[26] J. Hess, G. D. Tipaldi, and W. Burgard, "Null space optimization for effective coverage of 3d surfaces using redundant manipulators," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1923–1928.

[27] A. Behzad and M. Modarres, "A new efficient transformation of the generalized traveling salesman problem into traveling salesman problem," in *Proceedings of the 15th International Conference of Systems Engineering*, 2002, pp. 6–8.

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

[29] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotorsa modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.

[30] 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," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 598–626, 2015.

[31] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3d reconstruction," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3477–3484.