

Online Inspection Path Planning for Autonomous 3D Modeling using a Micro-Aerial Vehicle

Soohwan Song and Sungho Jo

Abstract—In this paper, we propose a novel algorithm for planning exploration paths to generate 3D models of unknown environments by using a micro-aerial vehicle (MAV). Our algorithm initially determines a next-best-view (NBV) that maximizes information gain and plans a collision-free path to reach the NBV. Along the path, the MAV explores the greatest unknown area although it sometimes misses minor unreconstructed region, such as a hole or a sparse surface. To cover such a region, we propose an online inspection algorithm that consistently provides an optimal coverage path toward the NBV in real time. The algorithm iteratively refines an inspection path according to the acquired information until the modeling of a specific local area is complete. We evaluated the proposed algorithm by comparing it with other state-of-the-art approaches through simulated experiments. The results show that our algorithm outperforms the other approaches in both exploration and 3D modeling scenarios.

I. INTRODUCTION

Autonomous 3D modeling of large environments by using mobile robots is one of the major research areas in computer vision and robotics. For this task, mobile robots consistently plan sensor configurations and construct 3D models of environments. In this study, we investigate a path planning problem of micro-aerial vehicles (MAVs) for constructing complete 3D models of unknown environments. MAVs are suitable for automatic modeling systems because of their high maneuverability and ability to reach almost any vantage point. However, owing to constrained power and payload, MAVs must be equipped with low capacity batteries and noisy vision sensors that experience short maximum ranges and limited fields of view. Consequently, an efficient path-planning algorithm is essential for their modeling systems.

This issue has been frequently addressed in mobile robot exploration. Most studies have been focused on the next-best-view (NBV) problem [1]. Their approaches consistently determine the most informative view by using feedback from the current partial reconstruction, and incrementally complete a model by sensing the view. The informativeness is defined through various approaches, such as the amount of unknown volumes [2] or frontiers [3] [4], a trend of reconstructed surfaces [5], or information-theoretic measures [6] [7]. Recently, some studies [8] [9] [10] attempted to determine the most informative sequence of views rather than a single optimal view. They computed optimal paths by evaluating a candidate set of view sequences and choosing the most informative sequence. All these approaches [2-10]

Soohwan Song and Sungho Jo are at Intelligent System and Neuromatics Lab (ISNL), School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Republic of Korea (e-mail: dramanet30@kaist.ac.kr, shjo@kaist.ac.kr)

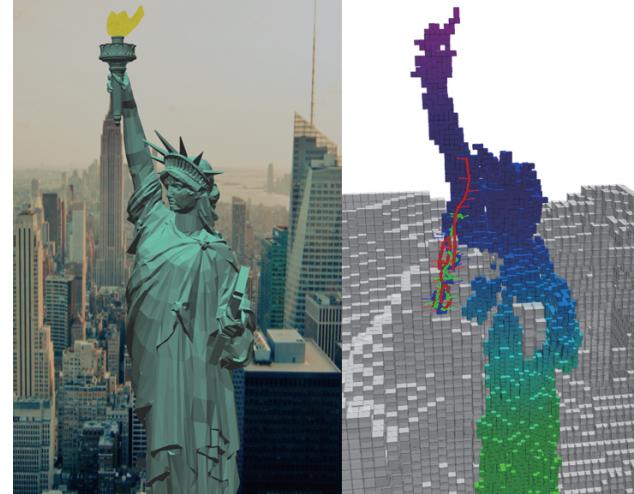


Fig. 1. Example of reconstructed 3D volumetric model (right) of the Statue of Liberty (left). The initially unknown space (gray) is updated as an MAV explores. The red, green and blue trajectories are inspection paths, which are sequentially refined by our online algorithm.

are greedy strategies that determine local solutions from current partial information. These greedy strategies focus only on reconstructing 3D environments that are extremely large, while ignoring less informative regions. Thus, the constructed model may be incomplete, and inaccurately reconstructed regions such as holes or sparse surfaces will be prevalent.

To address the aforementioned problem, we propose a novel path-planning algorithm that utilizes an inspection strategy to model an entire environment. Unlike other inspection problems [11] [12], which assume that prior map is known, we address online inspection according to partially known and consistently updated environments. Our algorithm first determines a goal configuration that maximizes information gain and computes an inspection path to the goal, providing a full coverage of frontiers. Next, the inspection path is iteratively refined according to the acquired information until modeling of a specific local area is complete. This online algorithm consistently provides an optimal inspection path in real time, guaranteeing maximum coverage and minimum path length. Fig. 1 depicts an example of constructed 3D volumetric model and inspection paths planned by our algorithm.

Summarily, this study makes the following contributions.
 i) Unlike past solutions, we apply an online inspection approach to the exploration problem to completely model a 3D environment. ii) We provide an efficient sampling strategy for

inspection-based path planning, which incrementally reduces the sampling range by utilizing a streaming set cover system [13]. iii) We empirically evaluate our approach in three simulated scenarios: the first is exploration of an office environment, the second involves modeling a single infrastructure, and the last involves modeling multiple infrastructures. The results prove the effectiveness and applicability of our method.

The remainder of this paper is structured as follows. Section II shows the related works on mobile robot exploration and inspection. Section III defines the considered problem, and Section IV describes the proposed solution in detail. Section V shows the experimental results of the proposed algorithm. Finally, Section VI summarizes the contributions made by this study.

II. RELATED WORKS

In this section, we discuss prior studies on mobile robot exploration and inspection in autonomous modeling systems. Approaches for the inspection task assume prior information about environments and plan a coverage path offline. In contrast, approaches for exploration do not assume any prior information and consistently plan paths online according to acquired information.

A. Exploration

As mentioned earlier, common exploration approaches for autonomous modeling systems involve greedy methods that iteratively determine an NBV and plan a collision-free path to reach the NBV. Yamauchi [3] was the first to introduce a frontier-based approach. He defined the frontier as a discrete boundary between the explored and unknown regions in a map. His approach involved planning a path toward the nearest frontier cells. Shen et al. [14] simulated the expansion of a particle system with Newtonian dynamics for 3D exploration of an MAV. The algorithm determines NBV as a region where a set of particles are significantly expanded. Vasquez-Gomez et al. [2] proposed a method that directly sampled candidate views in the configuration space. The NBV is determined by evaluating the samples by using a utility function.

Recently, studies have attempted to optimize paths toward the determined NBV. Dunn et al. [8] recursively searched intermediate paths to determine local optimal viewing configurations in a 2D space. Shade and Newman [9] computed a 3D vector field toward frontiers and determined the steepest descent path in the field. Bircher et al. [10] proposed an exploration algorithm based on a receding horizon NBV scheme. It constructs a rapidly exploring random tree (RRT) by sampling viewing configurations and accumulating the information gains along each branch. In addition, the method selects the first node of the best branch as a goal. Charrow et al. [15] found the most informative trajectory from global and local plans and refined it to maximize information gain. Even though this approach takes into account some locally uncertain parts of a map, it does not consider a full coverage of the local region. Therefore, this algorithm has

some possibility to miss some less informative regions of the map. Heng et al. [16] presented an exploration method that improves the coverage of unknown regions along the path to the NBV. This approach is similar to our algorithm because it utilizes an inspection approach for exploration of an MAV. However, their method requires a precomputation of 3D state lattices that contain motion-constrained edges and view frustums in every state whereas our approach operates in real time without any precomputation needed. The state lattices require a large amount of memory and computation in a huge and high-resolution map [17], so their approach [16] is not appropriate for precise modeling systems. Furthermore, it does not consider updated sensing information for the path, while our approach incrementally refines the inspection path according to updates.

B. Inspection

In the inspection-planning problem, a prior 3D model of the environment is required and inspection paths are planned offline. Englot and Hover [11] proposed a sampling-based approach to cover the complex 3D structures for inspecting a ship hull. The sampling-based approach separately solves the coverage-sampling problem and traveling salesman problem (TSP). The coverage-sampling problem aims to extract the smallest set of view configurations to guarantee full coverage. Further, the TSP aims to determine the shortest path to visit all sampled configurations. Papadopoulos et al. [12] proposed a random inspection-tree algorithm, considering differential vehicle constraints. The algorithm directly plans an inspection path in the configuration space by expending random trees. Galceran et al. [18] addressed an online inspection problem to handle state uncertainty for underwater vehicles. Similar to our method, the authors iteratively computed an inspection path according to sensing measurements. However, their system assumes that a prior model of the target is given, and computes initial paths offline by using the model. In contrast, our algorithm consistently provides an optimal inspection path without prior models in real-time and online.

III. PROBLEM DESCRIPTION

This study considers reconstructing an environment with an unknown but spatially bounded 3D space $V \subset \mathbb{R}^3$. We assume that dense 3D input data can be collected from a camera-based depth sensor. The sensor has several constraints such as limited field of view and maximum and minimum sensing ranges, which compose a view frustum [10]. The estimated 3D data is integrated into a probabilistic volumetric map based on an OctoMap [19]. The volumetric map \mathcal{M} simultaneously represents a workspace $W \subset \mathbb{R}^3$ and the volumetric model of environments. \mathcal{M} is composed of three states: occupied $V_{occupied} \subset V$, free $V_{free} \subset V$, and unknown $V_{unknown} \subset V$ spaces. Ultimately, our aim is to generate a complete volumetric model in which the percentage of unknown volume to the entire unknown volume in \mathcal{M} is lower than the threshold θ_{vol} in a short period.

Algorithm 1 Proposed path planning algorithm

Input: Volumetric map \mathcal{M} , Current configuration q_{curr} , Search distance d_{search} , and Initial sampling distance d_{sample} .

- 1: $[q_{goal}, \xi_{global}] \leftarrow ComputeGlobalPath(q_{curr})$
- 2: $R_{search} \leftarrow GetSearchRegion(\xi_{global}, d_{search})$
- 3: $queue_{Q^*} \leftarrow RandomSamples(\xi_{global})$
- 4: **while** $q_{curr} \neq q_{goal}$ **do**
- 5: $V_{front} \leftarrow GetFrontierCells(R_{search})$
- 6: **if** $\#(V_{front}^{new}) > \theta_{front}$ **then**
- 7: $[\xi_{local}, queue_{Q^*}] \leftarrow InspectionPathPlanning(q_{curr}, q_{goal}, queue_{Q^*}, V_{front}, d_{sample})$
- 8: **end if**
- 9: $MoveToward(\xi_{local})$
- 10: $Update(\mathcal{M}, q_{curr}, queue_{Q^*})$
- 11: **end while**

We assume that the configuration of an MAV is a flat state comprising a 3D position and yaw angle $q = \{x, y, z, \psi\}^T$, and the MAV's pose is already known. We denote the limit of translational speed as v_{max} , and the limit of rotational speed as $\dot{\psi}_{max}$. Both v_{max} and $\dot{\psi}_{max}$ are assumed to be small such that the roll and pitch angles are near zero. We plan paths only within the known free space V_{free} in \mathcal{M} , thus guaranteeing collision-free navigation. Let Q be a feasible configuration space, then a path $\xi : [0, 1] \rightarrow Q$ is defined as a sequence of configurations.

IV. PROPOSED APPROACH

To generate a 3D model of the environment, our approach iteratively plans an optimal path for an MAV by utilizing a current volumetric map \mathcal{M} until a desired model quality is achieved. Algorithm 1 explains an overview of the functioning of the proposed planning.

The algorithm first determines a goal configuration q_{goal} by evaluating a set of sampled view configurations, and choosing the view configuration with the highest information gain. Next, it computes a global path ξ_{global} , starting from the current configuration q_{curr} to the q_{goal} (line 1 and Section IV.A). Our algorithm iteratively plans a local path ξ_{local} providing the maximal coverage of the entire frontier cells near ξ_{global} . We define a search region $R_{search} \subset W$ as a set of positions inside radius d_{search} centered at each discretized position of ξ_{global} in \mathcal{M} (line 2). Our algorithm consistently updates the volumetric map \mathcal{M} (line 10) and frontier cells V_{front} inside R_{search} (line 5). We then iteratively plan the inspection path according to the updated frontier cells (line 7 and Section IV.B). In each iteration, our inspection algorithm performs an online refinement of the current local path ξ_{local} by maintaining a configuration set $Q^* \subset Q$. Q^* is a set of sampled configurations, which compose the local path ξ_{local} , and are sequentially stored in a queue structure $queue_{Q^*}$. The configurations that already passed by the MAV are removed from $queue_{Q^*}$ (line 10). The refinement step of the inspection path is performed only if the total number of new frontier cells $\#(V_{front}^{new})$ is greater than the constant

Algorithm 2 Online inspection path planning algorithm

Input: Current configuration q_{curr} , Goal configuration q_{goal} , Queue of configurations $queue_{Q^*}$, Frontier cells V_{front} , and Initial sampling distance d_{sample} .

- 1: $\xi_{short} \leftarrow ComputeShortPath(q_{curr}, q_{goal})$
- 2: $R_{sample} \leftarrow GetSamplingRegion(\xi_{short}, d_{sample})$
/* Online coverage sampling */
- 3: **for** $k = 1, \dots, N_{sample}^{local}$ **do**
- 4: $q_k \leftarrow GetFeasibleSample(queue_{Q^*}, R_{sample})$
- 5: $V_k \leftarrow Visible(q_k, V_{front})$
- 6: $T_k \leftarrow \operatorname{argmax}_{T_i \subset V_k} lev(T_i)$
- 7: **for all** $v_i \in T_k$ **do**
- 8: $eid(v_i) \leftarrow k$
- 9: $eff(v_i) \leftarrow lev(T_k)$
- 10: **end for**
- 11: $Q^* \leftarrow UpdateSampleSet(eid(\cdot))$
- 12: $d_{max} \leftarrow \max_{q_i \in Q^*} w_i$
- 13: **if** $(CovRatio(Q^*) > \theta_{cover}) \& (d_{max} < d_{sample})$ **then**
- 14: $d_{sample} \leftarrow d_{max}$
- 15: $R_{sample} \leftarrow GetSamplingRegion(\xi_{short}, d_{sample})$
- 16: **end if**
- 17: **end for**
- /* Solve TSP problem */
- 18: $\xi^* \leftarrow SolveTSP(\{q_{curr}, q_{goal}\} \cup Q^*)$
- 19: $[\xi^*, queue_{Q^*}] \leftarrow SmoothPath(\xi^*)$
- 20: **return** ξ^* and $queue_{Q^*}$

value θ_{front} (line 6). If the MAV reaches q_{goal} , we stop the iteration. These path-planning steps for goal selection and inspection are repeated until the model is completed.

A. Goal and Global Path Planning

To determine a goal, we first determine a sensor configuration q_{goal} that maximizes the information gain from the current partial reconstruction in \mathcal{M} . q_{goal} is defined as follows:

$$q_{goal} = \operatorname{argmax}_{k=1, \dots, N_{samples}^{global}} Gain(\mathcal{M}, q_k) \quad (1)$$

where $N_{samples}^{global}$ is the number of sampled configurations. The samples are directly generated in a feasible configuration space by extending branches of a rapidly exploring random tree star (RRT*) [20] from the current configuration q_{curr} . This approach, similar to that in [10], simultaneously processes the sample evaluation and path planning. Thus, all the samples in the (RRT*) are feasible to reach, and their paths are inherently collision-free.

The information gain $Gain(\mathcal{M}, q_k)$ is the number of unknown volumetric cells that can be observed at q_k , which is penalized by a distance factor [21]:

$$Gain(\mathcal{M}, q_k) = Vis(\mathcal{M}, q_k) \exp(-\lambda \cdot D(q_{curr}, q_k)) \quad (2)$$

where λ is a positive constant, and $D(q_{curr}, q_k)$ is the Euclidean distance of a collision-free path starting from q_{curr}

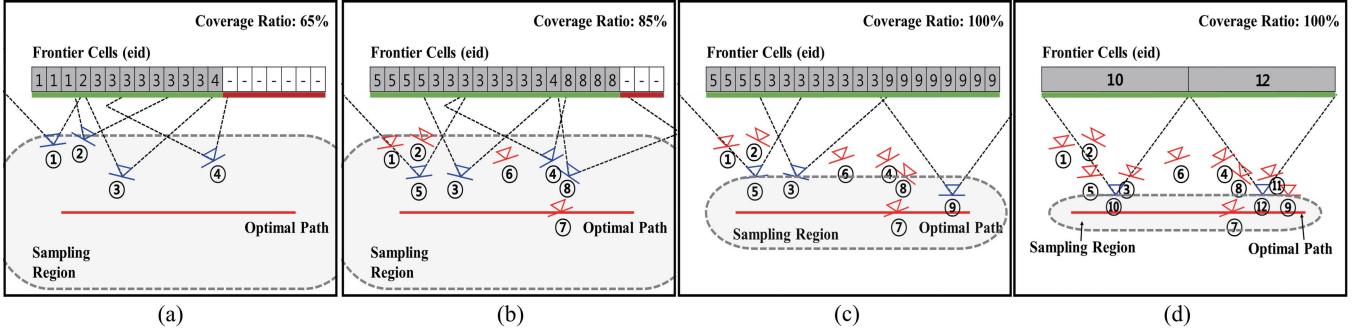


Fig. 2. An illustration of how our algorithm uses the streaming set cover approach [13] for incrementally reducing a sampling range and improving the coverage sampling problem. The processes are described in 2D for clarity. The red line and the gray ellipsoid represent the optimal path and sampling region, respectively. The successive boxes are frontier cells, and each box stores an identifier eid . Each configuration is sequentially sampled, starting from ① to ⑫. a) The configurations ① to ④ are assigned to each eid of frontier cell. b) The configurations ③ ⑤ ④ ⑧ are included in a suboptimal coverage set Q^* while ① ② ⑥ ⑦ are excluded. c) If the coverage ratio becomes 100%, our algorithm starts reducing the sampling region. d) Finally, we obtain the optimal coverage samples.

to q_k in the random tree. $Vis(\mathcal{M}, q_k)$ is the volume of visible and unknown cells from q_k in the current volumetric map \mathcal{M} . The volume is estimated through ray casting in the view frustum of the sensor and counting the number of unknown visible cells. After determining q_{goal} , we can obtain a global path ξ_{global} by extracting edges from q_{curr} to q_{goal} in the random tree. We then smoothed the path ξ_{global} by using a path smoothing method as in [22].

B. Online Inspection Path Planning

The proposed online inspection algorithm is detailed in Algorithm 2. We first compute the shortest path ξ_{short} from the current configuration q_{curr} to q_{goal} by using an RRT* planner [20] with path smoothing [22] (line 1). To prevent a situation in which the length of the inspection path is significantly longer than ξ_{short} , we restrict feasible sampling positions for path planning to a region R_{sample} . We define $R_{sample} \subset W$ as a set of positions in radius d_{sample} centered at each discretized position of ξ_{short} (line 2). As some frontier cells cannot be observed from positions in R_{sample} , we determine an optimal path that guarantees maximal coverage rather than full coverage.

To compute the optimal path, we employ a sampling-based approach [11] composed of a two-step optimization scheme. In the first step, the algorithm solves the coverage-sampling problem aimed to determine an optimal set of configurations that cover the frontier cells. In the second step, the algorithm solves the TSP, which aims to compute the shortest path connecting all sampled configurations.

To efficiently solve the coverage-sampling problem in real time, we employ a streaming set cover algorithm [13]. We represent the problem as a set system (V, Q) , where V is a finite set of frontier cells and Q is the robot configuration space. Every feasible configuration $q_k \in Q$ maps to a subset $V_k \subset V$ viewed by the sensor (line 5), and has its own weight w_k . When each configuration q_k is sampled individually, the goal is to construct a set of configurations $Q^* \subset Q$ that provides the maximum cover of V with the objective of minimizing the sum of their weights.

Our sampling algorithm repeatedly processes sampled configurations individually and outputs them for every frontier cell $v_i \in V_{front}$, an identifier $eid(v_i)$ of a sample q_k that representatively covers it, and an integer variable $eff(v_i)$ that intuitively captures the effectiveness of q_k in covering v_i . Thus, we consistently maintain a suboptimal coverage set Q^* online. To compute the effectiveness of a sample q_k , we define a level of a subset $T_k \subset V_k$ as

$$lev(T_k) = \frac{\#(T_k)}{\alpha \cdot w_k} \quad (3)$$

where α is a constant value, and $\#(T_k)$ is the number of elements in $v_i \in T_k$. w_k is defined as a proximity of q_k and ξ_{short} and can be computed as

$$w_k = \min_{q_i \in \xi_{short}} D(q_k, q_i) \quad (4)$$

where q_i is a discretized configuration in ξ_{short} . Subset T_k is said to be effective if for every $v_i \in T_k$, $lev(T_k) > eff(v_i)$. For each v_i in an effective set T_k , we assign the ID of sample q_k to $eid(v_i)$ and $lev(T_k)$ to $eff(v_i)$ (lines 7-10). If $lev(T_k) = 0$, sample q_k is eternally rejected.

The key advantage of this online set cover approach is that it consistently maintains a suboptimal solution Q^* in each iteration. We can estimate the sampling region of Q^* , which can be used to decrease the size of the sampling domain to possibly improve the solution. Thus, we can efficiently sample a configuration by incrementally reducing the sampling range R_{sample} . We first sequentially extract a sample q_k from $queue_{Q^*}$. After extracting all samples in $queue_{Q^*}$, we iteratively generate a uniform sample in R_{sample} (line 4). In each step, we compute a coverage ratio of Q^* . If the ratio is greater than the threshold θ_{cover} , we regard the samples as the first solution of the coverage sampling problem and start reducing R_{sample} (line 13). Let d_{max} be the maximum weight of a sample from Q^* , then set d_{sample} to d_{max} and recompute R_{sample} (lines 14 and 15). This approach finds more efficient samples that is close to the shortest path than the original coverage sampling algorithm [11]. Fig. 2 illustrates our coverage sampling approach.

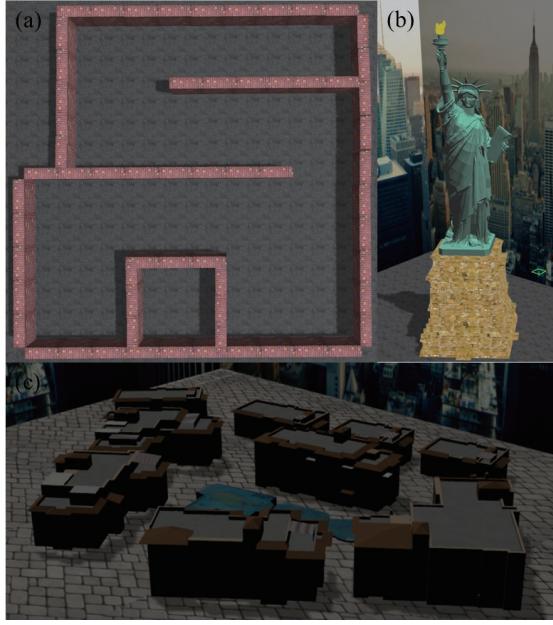


Fig. 3. Three simulated environments in ROS simulator: (a) office-like environment; (b) Statue of Liberty model; and (c) Village model.

TABLE I
PARAMETERS USED IN EXPERIMENTS.

Parameter	Scenario1	Scenario2	Scenario3
Map Size	$32 \times 32 \times 3 m^3$	$16 \times 16 \times 35 m^3$	$100 \times 70 \times 15 m^3$
Resolution of \mathcal{M}	$0.4m$	$0.2m$	$0.4m$
Field of View	$[60, 90]^\circ$	$[60, 90]^\circ$	$[60, 90]^\circ$
Max Range	$5m$	$8m$	$10m$
λ	0.5	0.5	0.5
d_{search}	$6m$	$10m$	$12m$
d_{sample}	$1m$	$2m$	$2m$
N_{sample}^{global}	5000	5000	5000
N_{sample}^{local}	1000	1000	1000
θ_{vol}	-	5%	10%
θ_{front}	20	100	200
θ_{cover}	90%	90%	90%

The final inspection path ξ^* is extracted by computing the shortest connecting path over all configurations $q \in Q^*$ by using a TSP solver [23] (line 18). We define a connection cost $cost(q_i, q_j)$ of the TSP solver as an execution time of motion [24]:

$$cost(q_i, q_j) = \max(D(q_i, q_j)/v_{max}, \|\psi_i - \psi_j\|/\dot{\psi}_{max}) \quad (5)$$

where $D(q_i, q_j)$ is the Euclidean distance directly connecting pairs q_i and q_j . If the connection has a collision, we use the RRT* planner to connect them. We then shorten the path length of ξ^* by using the heuristic speed-up improvement procedure in [25]. Each configuration in ξ^* is sequentially stored in $queue_{Q^*}$.

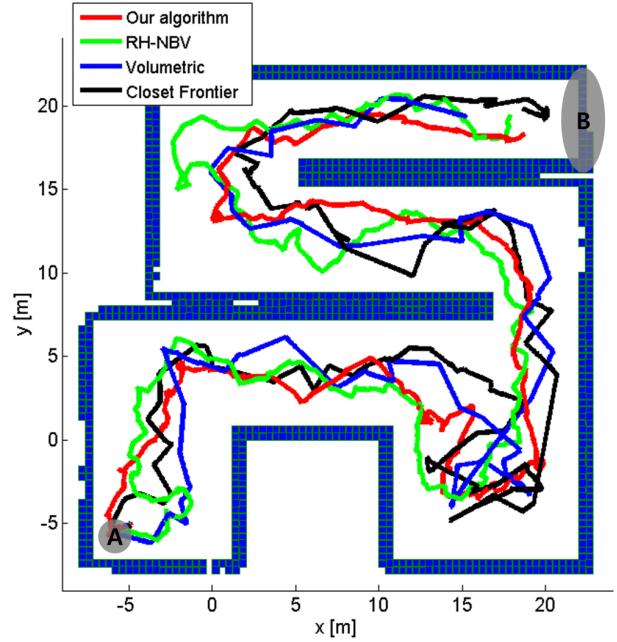


Fig. 4. Paths taken by the MAV using our algorithm, RH-NBV [10], volumetric approach [2], and closest frontier method [3] are colored red, green, blue, and black, respectively. Starting from region A, the MAV explores toward region B.

TABLE II
EXPERIMENTAL RESULTS IN SCENARIO1.

Algorithm	Completion Time(min)	Path Length(m)	Coverage (%)	Avg. Comp Time(sec)
Our Algorithm	27.32	145.05	93.84	0.31
RH-NBV [10]	28.45	153.25	93.69	0.20
Volumetric [2]	24.40	123.88	89.11	0.16
Closest Frontier [3]	30.07	160.67	92.47	0.14

V. EXPERIMENTAL RESULTS

In this section, we present the simulation experiments to evaluate the proposed approach. We employ the simulation system used in [10] that uses the model of a Firefly hexacopter MAV [26] in the RoterS simulation environment [27]. We consider three scenarios: i) exploring an office environment, ii) modeling a single infrastructure, and iii) modeling multiple infrastructures. Fig. 3 shows each simulation environment. To acquire 3D information of the simulated environment, we utilize a stereo camera mounted on the MAV with a downward pitch of 5° . The original camera specifications have a maximum range of $10m$, a minimum range of $0.3m$, and a field of view of $[60, 90]^\circ$. The maximum range for volumetric mapping is set according to the experimental situation for each test scenario. For the volumetric map \mathcal{M} , we set the probability interval for unknown cells to $[0.45, 0.55]$. Table I summarizes all parameters used in each scenario.

To evaluate the performance of our proposed approach, we

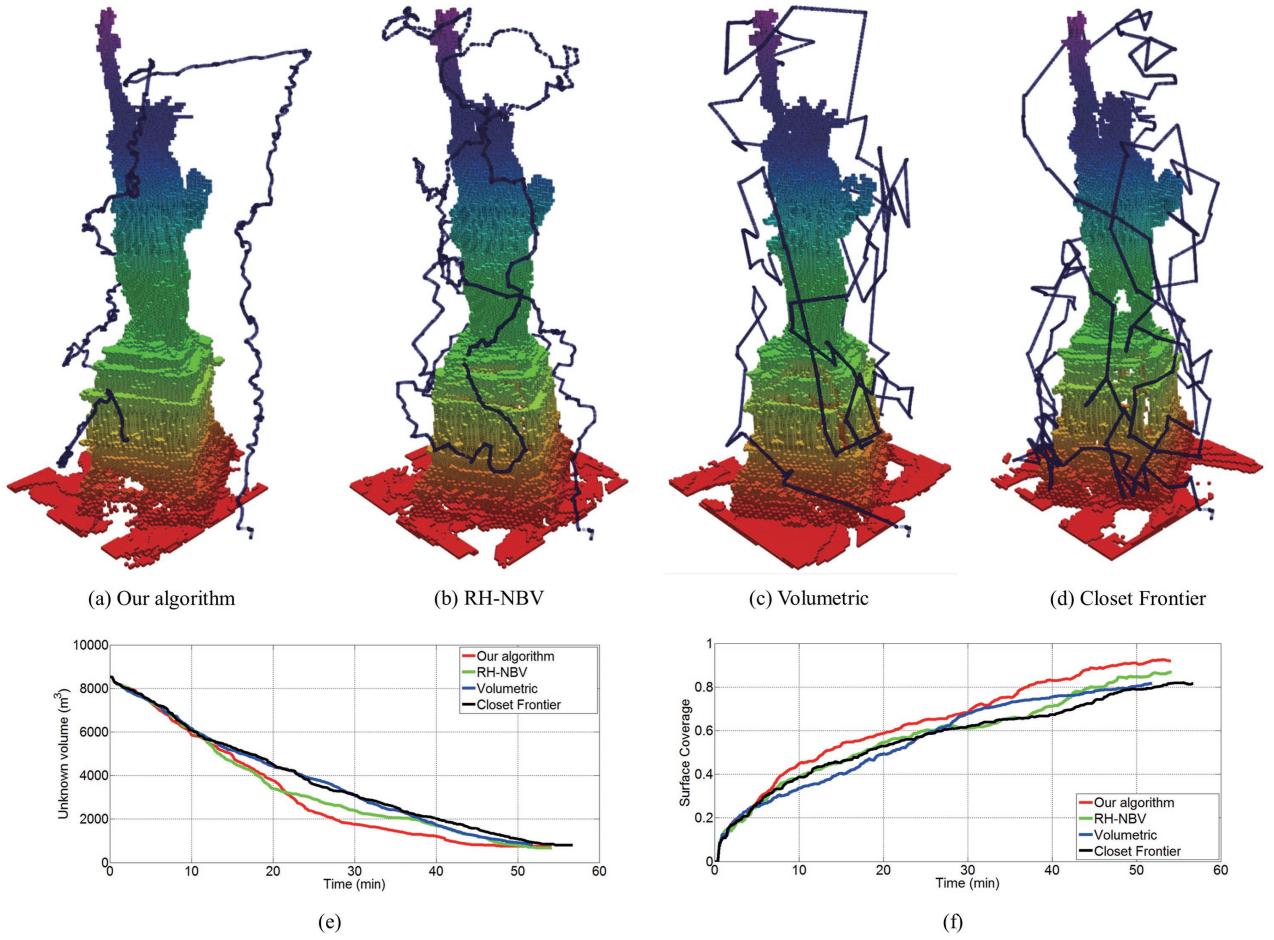


Fig. 5. Results in Scenario 2. Constructed volumetric models with resolution $0.2m$ of the Statue of Liberty object and trajectories taken by the MAV at the end of the executions of (a) our algorithm, (b) RH-NBV [10], (c) volumetric approach [2], and (d) closest frontier [3]. (e) Unknown volumes in the volumetric map, and (f) the surface coverage of the target over time.

compared our planning algorithm against three state-of-the-art approaches: the closest frontier approach [3], volumetric approach [2], and receding horizon NBV algorithm (RH-NBV) [10]. The closest frontier approach [3] has most commonly been used in 2D exploration applications. We extend the algorithm to 3D environments. We first cluster whole frontier cells, and then choose the closest cluster as NBV. The algorithm in [2] is similar to our global path-planning approach described in Section IV.A. The RH-NBV [10] is the latest exploration algorithm for autonomous modeling that evaluates an exploration path by directly expanding an RRT, and moves to the first edge of the best branch of the RRT. We ran all the algorithms on an Intel Core i7 CPU without a graphics-processing unit.

A. Exploration in office environment

In this experiment, volumetric maps of a structured and flat office environment are constructed using each exploration algorithm. Fig. 3a shows the structure of the office environment. Starting from region **A**, the MAV explores toward region **B**. When the MAV completely reconstructs the environment of region **B**, we stop the exploration and evaluate the performance. For each algorithm, we compute

the completion time of exploration, path length, percentage of covered volume, and average computation time. Table II shows the results, and each result is the average of 10 executions. Fig. 4 shows the trajectories taken by the MAV in each best case through different algorithms.

The volumetric approach has the best performance regarding completion time and path length. The volumetric approach generates the simplest and shortest path because it directly moves to the largest unknown region. However, it frequently misses small, unknown regions. Our algorithm shows the next best results in the completion time and path length, and shows the best coverage percentage. Compared to our algorithm, only the RH-NBV algorithm obtains a similar performance for the coverage percentage. The results of the average computation time show that all algorithms are operated in real time.

B. Modeling a single infrastructure

In this experiment, we evaluate the exploration performance with the modeling quality of each approach. The simulation environment consists of a target infrastructure (Statue of Liberty model [28]) placed on the ground within four walls (Fig. 3b). We assume that a bounded $16 \times$

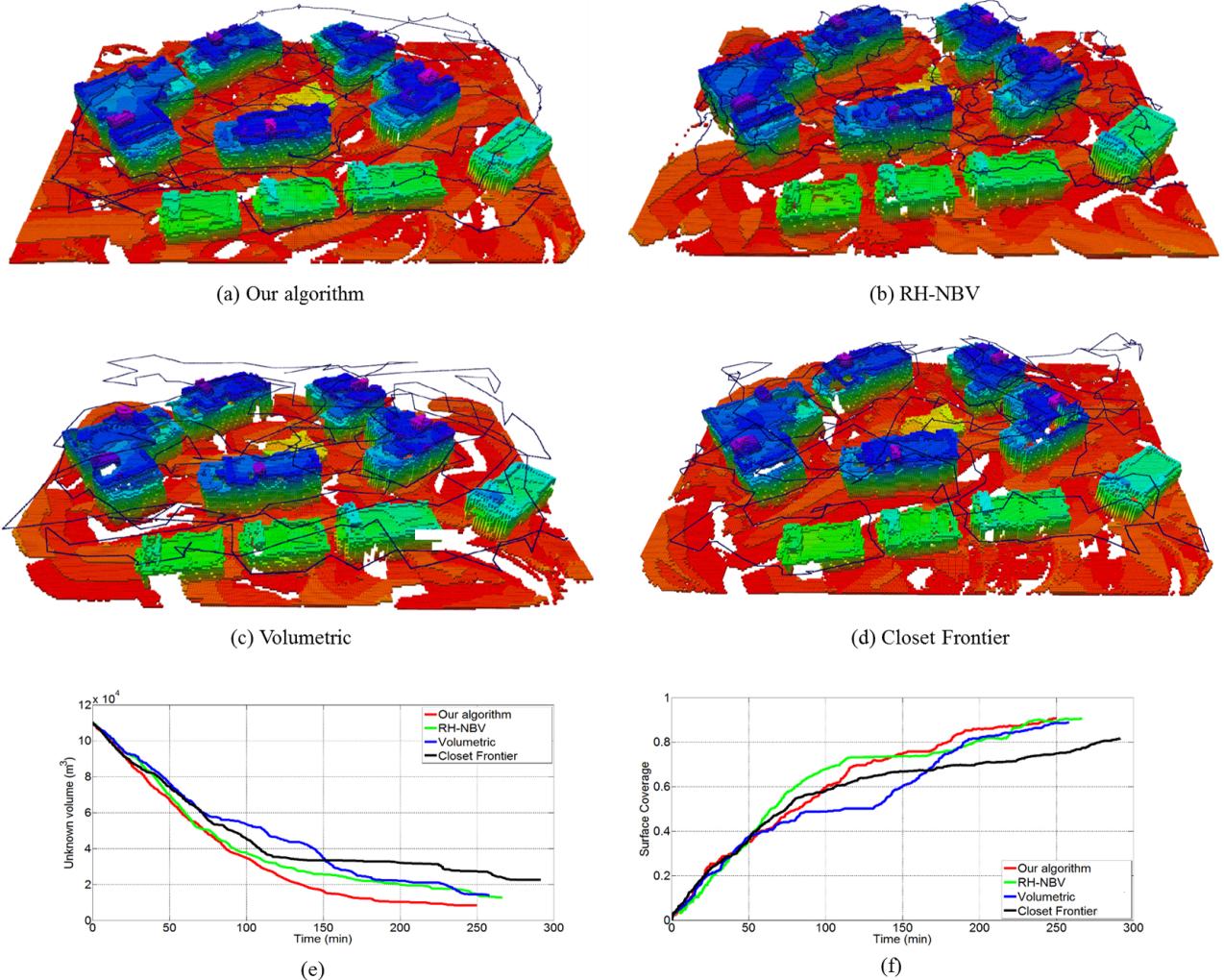


Fig. 6. Results in Scenario 3. Constructed volumetric maps of the Village model with resolution 0.4 m and trajectories taken by the MAV at the end of the executions of (a) our algorithm, (b) RH-NBV [10], (c) volumetric approach [2], and (d) closest frontier [3]. (e) Unknown volumes in the volumetric map, and (f) the surface coverage of the target over time.

$16 \times 35 m^3$ space containing the infrastructure is known and the remaining workspace of the robot is empty. The MAV explores the bounding space, while simultaneously constructing a volumetric map of the structure. To acquire an elaborate 3D model, we set the higher resolution of the map as $0.2 m$. For each algorithm, we perform 10 runs and average the results.

We present the results in Fig. 5. Figs. 5a–5d shows the paths and constructed models after the execution of each best case. The closest frontier method (Fig. 5d) frequently gets stuck in a local minimum. The volumetric approach (Fig. 5c) plans a broad path that evenly covers the entire area of the structure; however, the path sometimes overlaps. Fig. 5a shows that the path computed by our approach has a shorter trajectory length than other approaches, and simultaneously covers the entire area of the structure. Fig. 1 shows that the path constantly changes in yaw angle; this can improve the exploration performance and modeling quality.

Fig. 5e shows the unknown volume in the map over time.

It represents how fast each algorithm explores an unknown area. Fig. 5f shows the target's surface coverage, which is the percentage of observed surface cells compared to the total number of surface cells of the original model. It represents the completeness of the constructed volumetric model. The results show that our algorithm always outperforms the other algorithms. In particular, our algorithm has a surface coverage of 91.88%, indicating that our algorithm is very suitable for autonomous modeling systems.

C. Modeling multiple infrastructures

The third simulation environment is composed of nine buildings in a bounded $100 \times 70 \times 15 m^3$ space (Village model [28]). Starting at the side of the bounded space, the MAV explores the entire area until the percentage of the explored volume is greater than 90%. This scenario is closely related to real-world applications, such as search-and-rescue or environmental monitoring. As the amount of free volumes in this environment is much larger than the

occupied volumes, the result of the surface coverage is more significant than its progress.

Fig. 6 shows the results of this scenario demonstrating that our algorithm again performs the best. This indicates that our algorithm explores as much of the unknown volume as possible, and is simultaneously able to efficiently complete the 3D model.

VI. CONCLUSION

In this study, we proposed a novel algorithm for planning exploration paths of an MAV to construct 3D models of unknown environments. This algorithm first finds a goal configuration that maximizes information gain, and then iteratively plans an inspection path that provides the maximum coverage of neighboring frontier cells. To efficiently plan the inspection paths, we employ the streaming set cover approach [13], which helps reduce a sampling range of maximal coverage. Experimental results show that our algorithm performs better than other state-of-the-art algorithms and especially improves the completeness of the constructed volumetric models. To the best of our knowledge, this is the first work that implements an online inspection approach to autonomously explore and model environments using an MAV.

ACKNOWLEDGMENT

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

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Autonomous visual mapping and exploration with a micro aerial vehicle," *Journal of Field Robotics*, vol. 31, no. 4, pp. 654–675, 2014.
- [6] B. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1375–1392, 2014.
- [7] C. Forster, M. Pizzoli, and D. Scaramuzza, "Appearance-based active, monocular, dense reconstruction for micro aerial vehicle," in *2014 Robotics: Science and Systems Conference*, no. EPFL-CONF-203672, 2014.
- [8] E. Dunn, J. Van Den Berg, and J.-M. Frahm, "Developing visual sensing strategies through next best view planning," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 4001–4008.
- [9] R. Shade and P. Newman, "Choosing where to go: Complete 3d exploration with stereo," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2806–2811.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] Y. Emek and A. Rosen, "Semi-streaming set cover," *ACM Transactions on Algorithms*, vol. 13, no. 1, p. 6, 2016.
- [14] 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.
- [15] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, "Information-theoretic planning with trajectory optimization for dense 3d mapping," in *Robotics: Science and Systems*, 2015.
- [16] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys, "Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1071–1078.
- [17] 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.
- [18] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *Journal of Field Robotics*, vol. 32, no. 7, pp. 952–983, 2015.
- [19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [20] 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.
- [21] H. H. Gonzalez-Banos and J. C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.
- [22] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 995–1001.
- [23] 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.
- [24] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 6423–6430.
- [25] B. Englot and F. S. Hover, "Three-dimensional coverage planning for an underwater inspection robot," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1048–1073, 2013.
- [26] A. H. Autopilot, "Ascending technologies gmbh."
- [27] 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.
- [28] "Google's 3d warehouse." [Online]. Available: <http://3dwarehouse.sketchup.com/>