

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/335095224>

Coverage Path Planning using Path Primitive Sampling and Primitive Coverage Graph for Visual Inspection

Preprint · August 2019

CITATIONS

0

READS

97

5 authors, including:



Wei Jing

Carnegie Mellon University

29 PUBLICATIONS 107 CITATIONS

[SEE PROFILE](#)



Di Deng

Carnegie Mellon University

15 PUBLICATIONS 15 CITATIONS

[SEE PROFILE](#)



Zhe Xiao

Institute Of High Performance Computing

51 PUBLICATIONS 558 CITATIONS

[SEE PROFILE](#)



Kenji Shimada

Carnegie Mellon University

227 PUBLICATIONS 3,272 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Inspection [View project](#)



Architectural applications of my framework for anisotropic general patterns on building envelopes: roofs and facades. [View project](#)

Coverage Path Planning using Path Primitive Sampling and Primitive Coverage Graph for Visual Inspection

Wei Jing^{1,3,*}, Di Deng^{2,*}, Zhe Xiao³, Yong Liu^{1,3} and Kenji Shimada²

Abstract—Planning the path to gather the surface information of the target objects is crucial to improve the efficiency of and reduce the overall cost, for visual inspection applications with Unmanned Aerial Vehicles (UAVs). Coverage Path Planning (CPP) problem is often formulated for these inspection applications because of the coverage requirement. Traditionally, researchers usually plan and optimize the viewpoints to capture the surface information first, and then optimize the path to visit the selected viewpoints. In this paper, we propose a novel planning method to directly sample and plan the inspection path for a camera-equipped UAV to acquire visual and geometric information of the target structures as a video stream setting in complex 3D environment. The proposed planning method first generates via-points and path primitives around the target object by using sampling methods based on voxel dilation and subtraction. A novel Primitive Coverage Graph (PCG) is then proposed to encode the topological information, flying distances, and visibility information, with the sampled via-points and path primitives. Finally graph search is performed to find the resultant path in the PCG to complete the inspection task with the coverage requirements. The effectiveness of the proposed method is demonstrated through simulation and field tests in this paper.

I. INTRODUCTION

Recent years, vision-based structural inspection applications with Unmanned Aerial Vehicles (UAVs) in complex 3D environment have drawn increasingly attention from both industry and academia [1] [2] [3]. These inspection tasks require navigating a UAV around target objects to capture the visual information of the target surface area, for further investigation. The inspection is often a highly-repetitive task, and the target objects are in different sizes and geometries. Therefore, automatically generating efficient inspection paths for various target objects is crucial. Such planning task for inspection applications must satisfy surface coverage constraints, thus it is usually considered as a Coverage Path Planning (CPP) problem [4].

For the path planning of UAV visual inspection applications, the 3D geometric model of a target is available, or it can be built prior to the planning process. The change in target structure and ambient environment is often neglectable and could be considered as static. Therefore, the planning problem for visual inspection is considered as a

offline 3D coverage planning problem with model available. Given these application requirements, in this paper, we are interested in designing a offline, model-based coverage planning algorithm for structural inspection task with camera-equipped UAV.

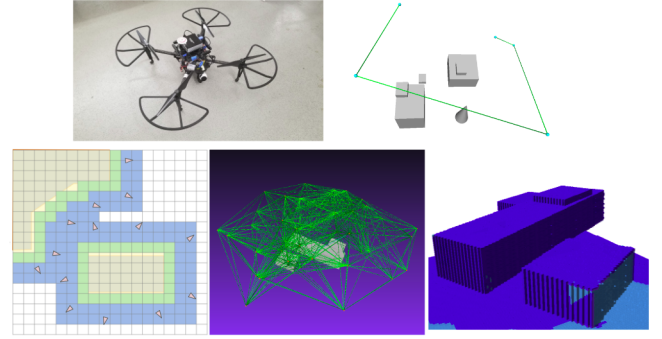


Fig. 1: Path Planning for Structural Inspection with Camera-equipped UAV

In this paper, we propose a novel computational planning method for CPP problem with application to the structure inspection with UAV, where the UAV is capturing visual information while flying around the target structure. The proposed method first generates via-points and path primitives around the target building; then the topological information, together with the traveling cost and visibility information, are used to build the Primitive Coverage Graph (PCG); after that, graph search method is performed on the PCG to find a inspection path that satisfies the application requirements. The main contributions of this paper are:

- a novel sampling-based coverage planning framework that is natively suitable for model-based, continuous CPP problem with UAV visual inspection, improved from our previous discrete viewpoint-based planning framework [1] [5].
- flexible and reconfigurable design of the planning framework such that the modules (e.g. sampling method, local planner, visibility model) could be substituted with different algorithms in order to make the framework adaptable for requirements from different applications.
- a novel sampling-based path primitive and via-point generation method for structural inspection application with UAVs, which allows direct search on the paths rather than the discrete viewpoints;
- a novel PCG-based strategy for the topology, cover-

¹ A*STAR Artificial Intelligence Initiative (A*AI); 1 Fusionopolis Way, 138632, Singapore.

² Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, 15213, USA.

³ Department of Computing Science, Institute of High Performance Computing; 1 Fusionopolis Way, 138632, Singapore.

* The authors equally contribute to the work, email address: jing.wei@ihpc.a-star.edu.sg, ddeng@andrew.cmu.edu

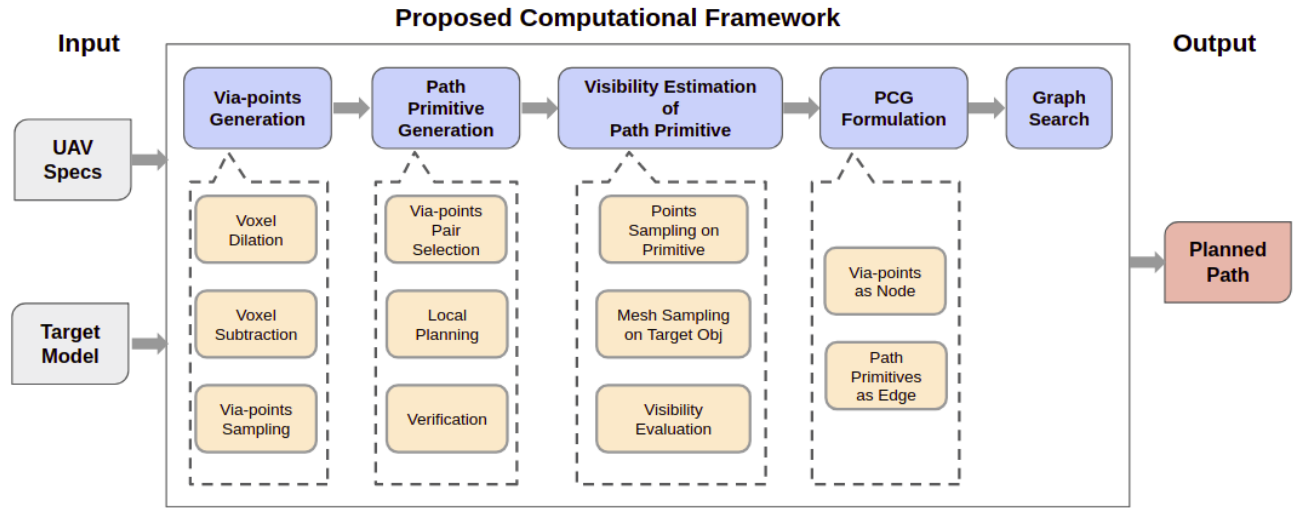


Fig. 2: The Block Diagram of the Proposed Coverage Planning Framework

age and cost information encoding and corresponding graph-search algorithms to solve the path finding-problem of the coverage planning.

II. RELEVANT WORK

Research work has been conducted on the CPP problems intensively in past years [2] [6] [7], because many real-world applications could be formulated as CPP problems.

Most of the previous work solves the CPP problem in two steps to address the requirement of visual information capturing. First, the planning algorithms identify a set of viewpoints that meet the coverage requirements for the inspection applications; the process of finding these viewpoints is called view planning or viewpoint planning problem (VPP) [8] [9]. Then the planning algorithms find the shortest path to maneuver a UAV/robot among the selected viewpoints to take the visual measurement, which is usually formulated as a (sequential) path planning problem [10] [11].

The sampling-based coverage planning method such as “Generate-Test” framework has been commonly applied to the model-based VPP in static, complex 3D environment, as summarized in the survey [8], where the methods sample a large set of candidate viewpoints, from which a subset of suitable viewpoints is selected according to the application requirements [7] [5] [12]. In addition, other view planning methods such as Next-Best-View (NBV) methods [13] [14], Reinforcement Learning based methods [15] and computational geometry based method [8] have also been used to address different requirements in different VPP applications.

In addition to viewpoint planning, the UAV/robot path is also required to be planned for the inspection applications. Early work [16] [11] formulated the path planning problem as a Traveling Salesman Problem (TSP) after finding the required viewpoints. This often yields sub-optimal results due to the separation of the optimization processes. Later Coverage Planning with combining viewpoint and path planning problem into one-step optimization problem has also been proposed to achieve better optimization results [17]

[18]. Additional previously planning methods for VPP and CPP problems in different applications could be found in several survey papers: [4] [9] [19] [8] [20] [21].

However, most of these methods focused mainly on planning and optimization of discrete viewpoints for visual information capturing, and then find a optimal/sub-optimal path to visit the selected viewpoints [16] [5] [22], or focused on coverage applications with continuous paths in simplified environment [23] [20]. Thus, these algorithms can not make full use of the video capturing features of the modern UAVs such that they are unsuitable or not performing well for our application. In this paper, instead of discrete viewpoints sampling and selection in the traditional sampling-based coverage planning methods [8] [1] [5], the proposed sampling-based coverage planning method directly samples the path primitives and then searches for a subset of the generated path primitive samples that satisfies the coverage requirements.

III. PROPOSED METHOD

In this paper, we propose a novel planning framework for the CPP problem of building inspection applications with UAV, using voxel dilation and subtraction, path primitive sampling, and Primitive Coverage Graph (PCG) encoding and searching. The overall proposed framework can be summarized as followings: The first step is to generate the via-points around the target buildings; then the path primitives are generated based on these sampled via-points; after that, the third step is to estimate the visibility information of the path primitives; then a PCG is generated to encode the topological information, traveling cost, and visibility information; the last step is to solve the inspection path-finding problem through graph search.

An overview of the proposed method is shown in Fig. 2. The computational framework is modularized and re-configurable such that different algorithms could be plugin to address different requirements for various applications, for example, different sampling strategies, local planners,

visibility models or search algorithms could be used in the corresponding modular to address the requirements from different applications.

A. Via-points Sampling by Voxel Dilation and Subtraction

In the proposed method, we first generate the via-points around the target structure in 3D space using the voxel-based sampling method, which utilizes importance sampling strategy to generate the via-points efficiently. This voxel-based via-points sampling method uses voxel dilation and subtraction to find the effective sampling region in the space. Our via-points generation method creates an initial feasible sampling region that satisfies both the safety and viewing-range requirements, scaling down the effective search of candidates for via-points, and accordingly facilitating the path primitive and PCG generation in later steps.

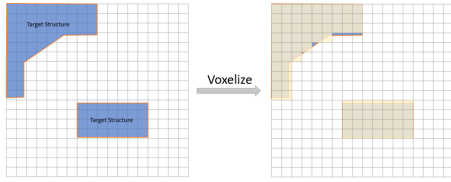


Fig. 3: Voxelizing the polygonal model of the target structure

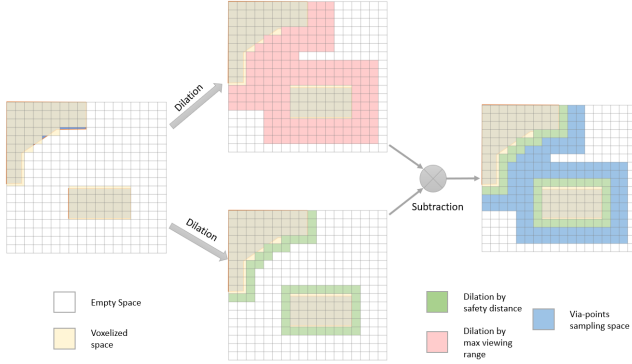


Fig. 4: Generating sampling region for via-points: the original voxel model is dilated twice, once by the safety offset (shown in green) and once more by the max viewing range (shown in red); after that, a subtraction operation is performed, making the resultant region the sampling region (shown in blue)

In the via-points generation step, the target structure is first voxelized to a 3D voxel model, as shown in Fig. 3. In order to compute efficient sampling region for the via-points, two voxel dilation operations are performed, one by the maximum viewing range and the other by a safety offset, as shown in Fig. 4; after that, voxel subtraction operation is performed between the two dilated voxel model to get the valid sampling region in the 3D Euclidean space; sampling of via-points is then performed by randomly choosing voxels in the sampling region, and then randomizing a position within the voxel. The overall process of the via-point generation is shown in Fig. 3 and 4.

In addition to the position, the viewing direction ν of a given via-point located at \mathbf{p}_{vp} is computed by the local potential field method [1], as shown in Eqn. 1:

$$\nu = \frac{\sum_i^N \frac{\mathbf{p}_{vp} - \mathbf{p}_{patch_i}}{\|\mathbf{p}_{vp} - \mathbf{p}_{patch_i}\|^3}}{\left\| \sum_i^N \frac{\mathbf{p}_{vp} - \mathbf{p}_{patch_i}}{\|\mathbf{p}_{vp} - \mathbf{p}_{patch_i}\|^3} \right\|}, \quad (1)$$

$$\text{for all: } \{ \mathbf{p}_{patch_i} | (\mathbf{p}_{vp} - \mathbf{p}_{patch_i})^T (\mathbf{p}_{vp} - \mathbf{p}_{patch_i}) < d_{max} \}, \quad (2)$$

where \mathbf{p}_{patch_i} is the position of the i^{th} surface patch; d_{max} is the maximum range of the local potential field.

B. Path-Primitive Sampling

Most previous ‘‘Generate-Test’’ methods consider the sampled via-points as candidate viewpoints; then select a subset from these viewpoints to fulfill the coverage requirements of the application; and finally plan the inspection paths to visit the selected viewpoints to complete the inspection task [16] [7] [5]. In this paper, however, we use these sampled points as via-points to generate path primitives, then directly select the path primitives as the final planning results. This approach is more intuitive and suitable to generate inspection path by considering the UAV inspection application as a continuous CPP problem in which a UAV continuously captures video stream whilst moving.

For the path primitive sampling process, we first choose the via-points pairs from the sampled via-points by randomly sampling pairs within a maximum distance; then a collision-free local planner \mathcal{L} is used to find a local path that connects the via-point pair, as shown in Fig. 5. In this paper, we directly connect two via-points with straight line as local planning process, with the collision check of the target objects. Other sampling-based planners such as Rapidly exploring Random Tree (RRT) [24] and Probabilistic Roadmap (PRM) [25] and their variations could also be used as local planner, for inspection applications such as robotic arms in more confined space to achieve better local planning performance.

An example of the sampled path primitives is illustrated in Fig. 7. The viewing direction of any point \mathbf{p}_t on the path primitives is computed by the interpolation of the viewing directions at the starting and ending via-points to ensure smooth transitions of angles while the UAV is moving, as shown in Eqn. 3.

$$\nu_t = \nu_s + \frac{\|\mathbf{p}_t - \mathbf{p}_s\|}{\|\mathbf{p}_e - \mathbf{p}_s\|} (\nu_e - \nu_s) \quad (3)$$

where ν_t and \mathbf{p}_t are the viewing direction and position at any point on the path primitive; $\nu_s, \mathbf{p}_s, \nu_e$ and \mathbf{p}_e are the viewing directions and positions at the starting and ending via-points, respectively.

Overall, given the planning parameters, the algorithm to sample the path primitives and the via-points is shown in Algo 1.

Algorithm 1 Path Primitive Sampling Algorithm

Input: The max distance for primitive sampling, d_{max} ; the max viewing range of sensor, d_{vis} ; the safety distance, d_{safe} ; the collision-free Local Planner, \mathcal{L} , the 3D model of target objects M ;

Output: The set of sampled via-points \mathbf{V} ; the set of sampled path primitives, \mathbf{E} ;

```

1:  $\mathbf{E}, \mathbf{V} \leftarrow \emptyset, \emptyset$ 
2:  $D_1 \leftarrow \text{dilate}(M, d_{vis})$ 
3:  $D_2 \leftarrow \text{dilate}(M, d_{safe})$ 
4:  $D \leftarrow \text{VoxelSubtract}(D_1, D_2)$ 
5: while  $\text{sizeof}(\mathbf{V}) < n_{vp}$  do
6:    $v \leftarrow \text{RandomSampleVP}(D)$ 
7:    $\mathbf{V} \leftarrow \text{append}(\mathbf{V}, v)$ 
8: end while
9: for  $v_i, v_j \in \mathbf{V}, i \neq j$  do
10:  if  $\text{dist}(v_i, v_j) \leq d_{max}$  then
11:     $e_{ij} \leftarrow \mathcal{L}(v_i, v_j, M)$ 
12:     $\mathbf{E} \leftarrow \text{Append}(\mathbf{E}, e_{ij})$ 
13:  end if
14: end for
15: return  $\mathbf{V}, \mathbf{E}$ 

```

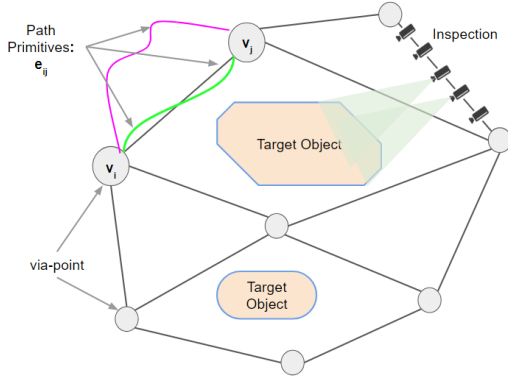


Fig. 5: Simplified 2D illustration of the path primitives and estimation of the visibility information

C. Visibility Estimation of Path Primitives

The visibility information is modeled as binary information that is pre-computed for each sampled path primitive, to avoid repetitive computation during the optimization process. Previous viewpoints-based approaches also pre-compute visibility, but only for individual viewpoint rather than path [7] [1].

In this paper, we propose a sampling-based method to estimate the visibility of the path primitive and the target building. The proposed visibility estimation method first samples viewpoints from the path primitives and estimate the binary visibility information of the surface patches and the sampled viewpoints, as shown in Fig. 5 and Fig. 6. The surface of the target object is subdivided to small and uniform triangular patches using the Bubble Mesh method [26], for the visibility evaluation. After estimating the visibility of the sampled viewpoints, we combine the visibility information from all sampled viewpoints on a path primitive to compute the overall visibility information of the path primitive.

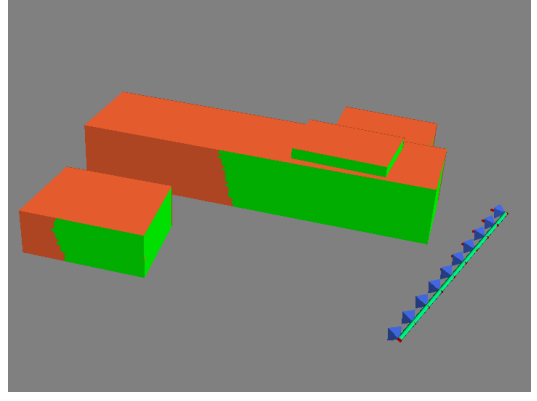


Fig. 6: Visibility estimation of path primitive, green areas are visible to the path primitive shown in the figure, computed with the visibility model. The viewpoints sampled on the path primitive are used to estimate the visibility, as shown in the figure

The visibility evaluation model between a individual sampled viewpoint and a surface patch is drawn from our previous work [5][1][3], and similar to [7][8], as listed below:

- The surface patch must be in the Field of View of the UAV from the sampled viewpoint.
- The surface patch must be within the viewing range of the UAV from the sampled viewpoint.
- The viewing angle must be within a certain range predicted by the sensor specifications.
- There must be no occlusion between the sampled viewpoint and surface patch.

Then the visibility information for a path primitive is combined from the sampled viewpoints and represented in a $m \times 1$ binary vector \mathbf{s} , where m is the number of surface patches of the target structure. A 3D visualization results of visibility estimation of a given path primitive is shown in Fig. 6. It will be associated with the corresponding edge and stored in the Primitive Coverage Graph (PCG).

D. Primitive Coverage Graph for Information Encoding

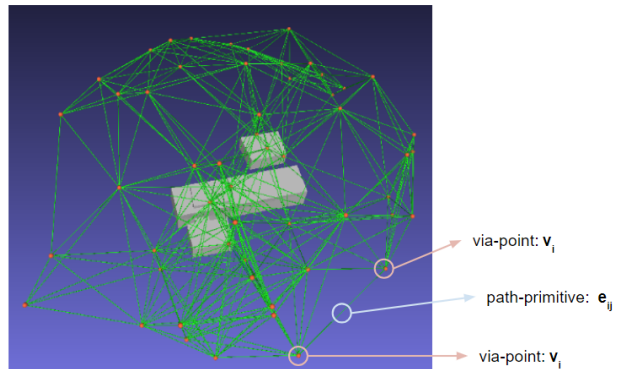


Fig. 7: Illustration of sampled path primitives and via-points in 3D space (this figure is for visualization, the via-points and path primitives are much denser in the implementation)

In this paper, we propose a novel Primitive Coverage Graph (PCG) representation $\mathcal{G}(\mathbf{V}, \mathbf{E})$ to encode the topological information, UAV flying distance (traveling cost) as well as visibility information of the sampled via-points and path primitives. As visualized in Fig. 7, node $v_i \in \mathbf{V}$ is a via-point and edge $e_{ij} \in \mathbf{E}$ represents a path primitive that connects via-points v_i and v_j .

For the PCG proposed in this paper, the key information is associated with the edges \mathbf{E} that correspond to the sampled path primitives. The nodes \mathbf{V} encode the topological information between the path primitives, for the assurance of the connectivity and no-branching of the final path in the graph search in later step. The information associated with edge e_{ij} is $\{d_{ij}, \mathbf{s}_{ij}\}$, where d_{ij} is the length of the path primitive; \mathbf{s}_{ij} is a $m \times 1$ vector to model the binary visibility information of sampled patches of the target structure. The visibility information of a path primitive is computed in Sect. III-C. For element $s_{ij}^k \in \mathbf{s}_{ij}, k \in [1, m]$, $s_{ij}^k = 1$ indicates a surface patch indexed by k is visible to path primitive e_{ij} , and $s_{ij}^k = 0$ indicates it is invisible. The PCG representation and information encoding not only ensures the graph search step is able to find a fully connected path without branches, but also ensures that the coverage constraints are satisfied.

The PCG construction methods are summarized in Algo. 2. It takes the sampled via-points \mathbf{V} and the sampled path primitives \mathbf{E} from Algo. 1 as input, and generate the PCG $\mathcal{G}(\mathbf{V}, \mathbf{E})$ as output to encode the information.

Algorithm 2 Primitive Coverage Graph Construction Algorithm

Input: The 3D model of target objects, \mathbf{M} , the set of sampled via-points, \mathbf{V} ; the set of sampled path primitives, \mathbf{E} ;

Output: The Primitive Coverage Graph, $\mathcal{G}(\mathbf{V}, \mathbf{E})$;

```

1: for  $e_{ij} \in \mathbf{E}$  do
2:    $\mathbf{s}_{ij} \leftarrow \text{VisibilityEstimation}(e_{ij}, \mathbf{M})$ 
3:    $d_{ij} \leftarrow \text{ComputeDist}(e_{ij})$ 
4:    $e_{ij} \leftarrow \text{Associate}(e_{ij}, \mathbf{s}_{ij}, d_{ij})$ 
5: end for
6:  $\mathcal{G}(\mathbf{V}, \mathbf{E}) \leftarrow \text{BuildGraph}(\mathbf{V}, \mathbf{E})$ 
7: return  $\mathcal{G}(\mathbf{V}, \mathbf{E})$ 

```

where *VisibilityEstimation* described in Sect. III-C is to estimate the visibility information of each sampled path primitive.

E. Primitive Coverage Graph Search

The path finding problem is formulated with the PCG, $\mathcal{G}(\mathbf{V}, \mathbf{E})$, that encodes the topological information, traveling distances and coverage information of the path primitives. The objective of the path planning problem is to find a fully connected, branch-less path from the PCG that minimize the traveling distance while the required coverage ratio δ_d is achieved.

In this paper, we propose a Greedy Neighborhood Search (GNS) algorithm to search the final path directly for the CPP problem, with all the information encoded in the PCG. The details of the algorithm is shown in Algo. 3.

Algorithm 3 Greedy Neighborhood Search Algorithm for Path Finding on PCG

Input: The desired coverage ratio, δ_d ; the Primitive Coverage Graph, $\mathcal{G}(\mathbf{V}, \mathbf{E})$, and other algorithm specific parameters;

Output: the resultant path primitives \mathbf{E}_{res} ;

```

1:  $\mathbf{V}_{res}, \mathbf{E}_{res}, \delta \leftarrow \emptyset, \emptyset, 0$ 
2:  $\delta \leftarrow \text{FindCoverage}(\mathcal{G}, e)$ 
3:  $v, e \leftarrow \text{Initialize}(\mathcal{G})$ 
4: while  $\delta \leq \delta_d$  do
5:    $\mathbf{E}_{nbr} \leftarrow \text{FindNeighbour}(v, \mathcal{G})$ 
6:    $v, e \leftarrow \text{FindTraj}(\mathbf{E}_{nbr}, \mathbf{E}_{res}, \mathcal{G})$ 
7:    $\mathbf{V}_{res} \leftarrow \text{Append}(\mathbf{V}_{res}, v)$ 
8:    $\mathbf{E}_{res} \leftarrow \text{Append}(\mathbf{E}_{res}, e)$ 
9:    $\delta \leftarrow \text{FindCoverage}(\mathcal{G}, \mathbf{E}_{res})$ 
10: end while
11: return  $\mathbf{E}_{res}$ 

```

For the proposed GNS algorithm, at each iteration in the loop, the algorithm searches the current neighborhood of the path primitives that yields highest incremental coverage over distance. The path primitive will be added to the solution set \mathbf{E}_{res} that consists of a number of path primitives with order. Then the coverage information and neighbourhood will be updated. The iteration will repeat until the desired coverage ratio, δ_d , is achieved.

The *FindTraj* in GNS is to find the path primitive in the neighbourhood, based on the incremental coverage over distance, given the current via-point and path primitives, as shown in Eqn. 4.

$$e = \underset{e \in \mathbf{E}_{nbr}}{\operatorname{argmax}} \frac{\Delta \text{cov}(e, \mathbf{E}_{res})}{d_e} \quad (4)$$

where d_e is the flying distance of the path primitive associated with edge e ; \mathbf{E}_{nbr} is the neighbourhood edges (the edges connected to current via-point v), identified by *FindNeighbour* using the topological information in \mathcal{G} ; \mathbf{E}_{res} is the current set of edges in the resultant path, which provides the total current coverage information; $\Delta \text{cov}(e, \mathbf{E}_{res})$ is the incremental surface coverage by the path primitive, e .

IV. RESULTS

A. Setup

In this paper, simulation and experimental tests are conducted to validate the effectiveness of the proposed planning method. In the simulation tests, we first applied the proposed planning method to compute the inspection paths; then the drone is flied on the planned paths in Drake [27], a toolbox for robotic simulation and analysis, to capture the visual information; after that the Octomap library [28] is used to build the 3D voxel model of the target structure to validate inspection results of the computed UAV paths. In the real-world experiment, a DJI drone is used to capture the required surface information with the planned paths, and the 3D surface of the target objects is reproduced to further demonstrate the effectiveness of the proposed method.

B. Simulation Tests with Drake and Octomap

1) *Planning Parameters and Results:* First, the proposed method is used to compute the inspection path of the UAV for the two different target buildings. 1,765 and 1,410 path primitives are generated around each target structure, respectively. The parameters used for the coverage planning process is shown in Table I. The resultant paths computed by the proposed method are visualized in Figs. 8a and 8b.

TABLE I: Planning parameters used in this paper

Parameter	Value
FOV (Diagonal)	94°
Max Viewing Angle	75°
Resolution (Pixels)	4000 × 3000
Coverage Ratio (δ_d)	99.0%
Safety Distance	2m
Max Viewing Range	50m

The result comparison with two previous methods is shown in Table II, based on the average of 10 trials. The flying distances computed by the proposed method in this paper are 425.6m and 466.2m for the two target structures, respectively. The distances of the inspection paths are 507.7m and 587.5m using the previous methods [5][11] that solves the TSP after applying the viewpoint-based methods to find the viewpoints (labeled as VPP-TSP). The discrete viewpoints based greedy method, a simple but robust baseline [15][11] that iteratively adds the most cost effective viewpoints, planned the path with the traveling distance of 531.0m and 687.1m. Assuming a constant flying speed, the reductions of the required inspection time are 18.4% and 29.2% on average, compared to the two previous methods, respectively.

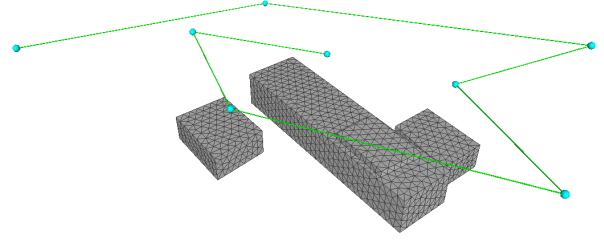
TABLE II: Planning results

Methods	Structure 1	Structure 2
The Proposed Method	425.6m	466.2m
VPP-TSP[5][11]	507.7m	587.5m
Greedy Method	531.0m	687.1m

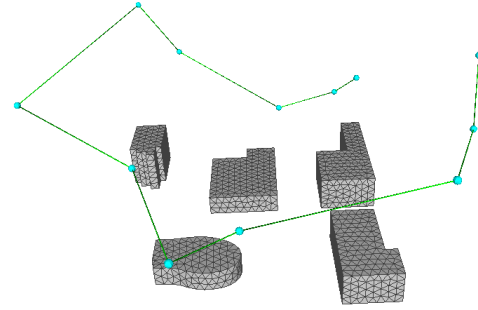
2) *Computational Simulation Tests:* Simulation tests implemented with Drake simulator [27] are performed to validate the coverage and effectiveness of the proposed method. Fig. 9 shows the simulation test work-flow. With the camera simulator provided by Drake, point cloud data is generated based on the camera poses. The generated point cloud data in each frame is located based on the world frame through Eqn. 5:

$$\mathbf{P}^{world} = \mathbf{T}_{cam}^{world} \mathbf{P}^{cam}, \quad (5)$$

where \mathbf{P}^{world} is the position of the point cloud data in the world frame, \mathbf{T}_{cam}^{world} is the pose of the camera in the world frame and \mathbf{P}^{cam} is the position of the point cloud with respect to the camera frame. The Octomap library [28] is then used to create the occupancy voxel models from the point cloud results. The visualization of the reconstructed voxel models is shown in Fig. 10.



(a) Target Structure 1



(b) Target Structure 2

Fig. 8: Visualization of planned UAV inspection path around the target buildings

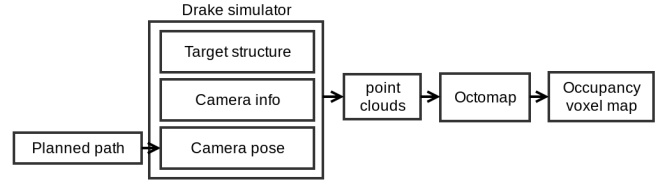


Fig. 9: Simulation Flowchart with Drake and Octomap

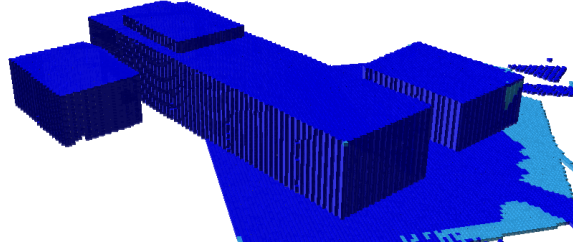
C. Field Test

An field test has also been conducted to further validate the proposed planning method in real-world scenario. In the field test, a DJI Matrice 100 is used to fly around the target structure to capture the surface information, with the path computed by the proposed method. The max viewing range used for this experiment is 10m, in order to get better details for reconstructions for in practical applications; the safety distance used for the experimental test is 0.5m. The planned path is shown in Fig. 11.

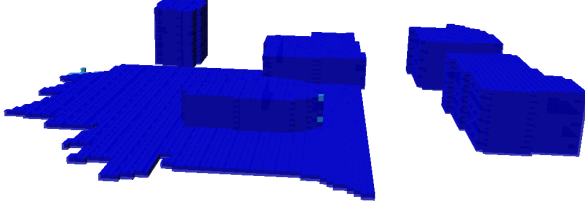
The 3D visualization of the resultant reconstructed model is shown in Figs. 12 and 13. The reconstruction is done by using Structure From Motion (SFM) method with 87 pictures sampled from the video taken by the drone, during the inspection process. VisualSFM [29] and CMP-MVS [30] are used in this papaer for SFM and surface reconstruction.

V. DISCUSSION

The proposed planning method is shown to be able to automatic generate efficient path for different target objects



(a) Target Structure 1



(b) Target Structure 2

Fig. 10: Voxel model of results from Drake-Octomap simulation using the planned paths with the proposed algorithm

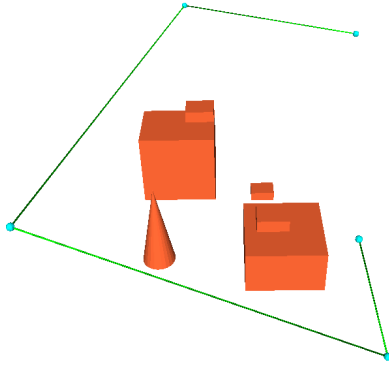


Fig. 11: Visualization of the Planned Path for the Experiment

in 3D environment, with the coverage requirements. Both simulation and field tests validate that the proposed planning method successfully covered the surface area of the target objects.

Compared to previous discrete viewpoints sampling-based methods [1] [7], the proposed path primitive sampling method is able to find shorter path for the UAV inspection tasks, which helps to improve the efficiency and reduces the cost. The proposed method targets for the modern UAV system with high performance of continuous video capturing features, which is more naturally suitable for these applications, while the viewpoints sampling based method requires the UAV to take pictures at discrete viewpoints. Moreover, the discrete viewpoint sampling method is known to be unreliable for registration [11], because the viewing angles from different viewpoints may cause failure in image



Fig. 12: Visualization of 3D reconstruction results of the field tests

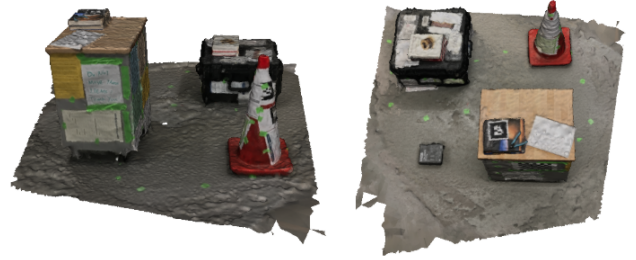


Fig. 13: Additional views of 3D reconstruction results of the field tests

registration even if there are enough overlap areas of two images. For path primitive sampling method, however, the registration problem is usually not a concern because the images are sampled from continuous motion.

Additionally, the proposed planning framework is flexible and reconfigurable. Different algorithms/modules could be used to replace the current ones such that the proposed framework is adoptable for other applications, for example, product shape inspection problem with robotic manipulator [31] or mobile robot [13]. Moreover, with properly modeling of the process, it is also possible to extend the proposed planning framework to other planning problem in robotic applications with coverage constraints, such as robotized painting, surface polishing.

VI. CONCLUSION

In this paper, we presented a novel path-primitive based, offline planning framework for the continuous CPP problem, with applications of surface inspection using UAV. The proposed method utilizes voxel-based via-points sampling, path primitive sampling, PCG encoding, and graph search to plan UAV inspection paths that satisfy the coverage requirements. We also demonstrated the effectiveness of the proposed framework and the implementation in both simulation and experimental tests with different target objects. Through the

comparison with previous methods, the reductions on the required inspection time are found to be 18.4% and 29.2% on average.

ACKNOWLEDGEMENT

This research is supported by the Agency for Science, Technology and Research (A*STAR), Singapore, under its AME Programmatic Funding Scheme (Project #A18A2b0046).

REFERENCES

- [1] W. Jing, J. Polden, W. Lin, and K. Shimada, "Sampling-based view planning for 3d visual coverage task with unmanned aerial vehicle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2016, pp. 1808–1815.
- [2] 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 *IEEE International Conference on Robotics and Automation*. IEEE, 2015, pp. 6423–6430.
- [3] D. Deng, T. Pang, P. Palli, F. Shu, and K. Shimada, "Heterogeneous vehicles routing for water canal damage assessment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2018, pp. 2375–2382.
- [4] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [5] W. Jing, J. Polden, P. Y. Tao, W. Lin, and K. Shimada, "View planning for 3d shape reconstruction of buildings with unmanned aerial vehicles," in *International Conference on Control, Automation, Robotics and Vision*. IEEE, 2016, pp. 1–6.
- [6] 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.
- [7] W. R. Scott, "Model-based view planning," *Machine Vision and Applications*, vol. 20, no. 1, pp. 47–69, 2009.
- [8] W. Scott, G. Roth, and J.-F. Rivest, "View planning for automated 3d object reconstruction inspection," *ACM Computing Surveys*, vol. 35, no. 1, 2003.
- [9] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.
- [10] S. Alartartsev, S. Stellmacher, and F. Ortmeier, "Robotic task sequencing problem: A survey," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 2, pp. 279–298, 2015.
- [11] W. Jing, "Coverage planning for robotic vision applications in complex 3d environment," Ph.D. dissertation, Carnegie Mellon University, 2017.
- [12] G. H. Tarbox and S. N. Gottschlich, "Planning for complete sensor coverage in inspection," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 84–111, 1995.
- [13] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View planning for 3d object reconstruction with a mobile manipulator robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4227–4233.
- [14] S. Song and S. Jo, "Online inspection path planning for autonomous 3d modeling using a micro-aerial vehicle," in *IEEE International Conference on Robotics and Automation*. IEEE, 2017, pp. 6217–6224.
- [15] M. D. Kaba, M. G. Uzunbas, and S.-N. Lim, "A reinforcement learning approach to the view planning problem," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5094–5102.
- [16] S. Chen and Y. Li, "Automatic sensor placement for model-based robot vision," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 393–408, 2004.
- [17] W. Jing, J. Polden, C. F. Goh, M. Rajaraman, W. Lin, and K. Shimada, "Sampling-based coverage motion planning for industrial inspection application with redundant robotic system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017, pp. 5211–5218.
- [18] P. Wang, R. Krishnamurti, and K. Gupta, "View planning problem with combined view and traveling cost," in *IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 711–716.
- [19] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, and G. Cai, "A survey on inspecting structures using robotic systems," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, p. 1729881416663664, 2016.
- [20] H. Choset, "Coverage for robotics—a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1–4, pp. 113–126, 2001.
- [21] K. A. Tarabani, P. K. Allen, and R. Y. Tsai, "A survey of sensor planning in computer vision," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 86–104, 1995.
- [22] M. Roberts, S. Shah, D. Dey, A. Truong, S. N. Sinha, A. Kapoor, P. Hanrahan, and N. Joshi, "Submodular trajectory optimization for aerial 3d scanning," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 5334–5343.
- [23] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Asymptotically optimal inspection planning using systems with differential constraints," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4126–4133.
- [24] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [25] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 12, no. 4, 1996.
- [26] K. Shimada and D. C. Gossard, "Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing," in *Proceedings of the third ACM symposium on Solid modeling and applications*. ACM, 1995, pp. 409–419.
- [27] R. Tedrake and the Drake Development Team, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2016. [Online]. Available: <https://drake.mit.edu>
- [28] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at <http://octomap.github.com>. [Online]. Available: <http://octomap.github.com>
- [29] C. Wu, "VisualSFM: A visual structure from motion system," *URL http://ccwu.me/vsfm/*, 2011.
- [30] M. Jancosek and T. Pajdla, "Multi-view reconstruction preserving weakly-supported surfaces," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 3121–3128.
- [31] W. Jing, J. Polden, P. Y. Tao, C. F. Goh, W. Lin, and K. Shimada, "Model-based coverage motion planning for industrial 3d shape inspection applications," in *IEEE Conference on Automation Science and Engineering*. IEEE, 2017, pp. 1293–1300.