2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Sampling-Based View Planning for 3D Visual Coverage Task with Unmanned Aerial Vehicle

Wei Jing[1], Joseph Polden[2], Wei Lin[2] and Kenji Shimada[1]

Abstract— **The view planning problem is the problem that involves finding suitable viewpoints for vision-related tasks such as inspection or reconstruction. In this paper, we propose a novel view planning algorithm for a camera-equipped Unmanned Aerial Vehicle (UAV) acquiring visual geometric information of target objects in its surrounding environment. The proposed model-based approach makes use of iterative random sampling and a probabilistic potential-field method to generate candidate viewpoints in a non-deterministic manner. Combinatorial optimization is then applied to select the most suitable subset of these candidate viewpoints to complete the given visual inspection or shape reconstruction task. The effectiveness of the proposed method is demonstrated through a number of computational tests that compare its overall performance against two previous methods. A field-test is also performed to demonstrate the method's applicability in a real world UAV-based shape reconstruction task of an outdoor statue.**

I. INTRODUCTION

Nowadays, many commercial Unmanned Aerial Vehicles (UAVs) are equipped with an on-board camera that can be used for vision tasks such as visual inspection [1] and shape reconstruction [2]. These visual tasks require navigating the UAV around a target object whilst capturing a series of photographs which cover the surface area of the entire target object, or a designated portion of it. Shape reconstruction tasks are more demanding; each section of the target surface area must be captured at least twice from different positions to ensure a full 3D model of the target object can be reconstructed via triangulation.

For both inspection and reconstruction, the first step is to identify a set of viewpoints which meet the required goal(s); the process of finding these viewpoints is called view planning [3][4][5]. The ideal set of viewpoints would consist of a small number of viewpoints with high overall coverage ratio of the target surface. Performing these view planning tasks with acceptable computational cost is also an important consideration.

During past few years, a significant body of work in view planning for different applications has been published [2][6][7]. In the 1990s, the early work focussed mostly on view planning for small target objects such as mechanical parts. A number of these early approaches made use of computational geometry [3], these approaches were found to suffer from high computational cost when more complex target geometry was involved.

Later "generate-and-test" methods became popular as summarized in [4]. These methods generate a large set of initial candidate viewpoints, from which a subset of suitable viewpoints is selected through combinatorial optimization [4][8]. Candidate viewpoints are typically generated by offsetting from the target objects surface [7], or on the surface of a sphere which encapsulates it [9]. These methods worked well on target objects with relatively simple geometry. However, since candidate viewpoints generated by these methods are sampled from a small fraction of all feasible viewpoints, the number of resultant viewpoints may be high, and the coverage ratio may be poor for large complex target objects.

In more recent years, view planning has been incorporated into UAV applications where large target objects such as buildings or outdoor environments need to be inspected or reconstructed via aerial photography [10][2]. Most of the work, however, focussed on partial coverage problems of view planning [10], or in the adaptation of offsetting methods to generate candidate viewpoints [2]. These characteristics limited the performance of these methods for the view planning problem discussed in this paper.

In this paper, a new iterative two-step sampling-based approach is proposed for view planning problem. The proposed method firstly generates candidate viewpoint positions around the target object by random sampling, and then generates the viewing directions for each of these positions with a novel probabilistic potential-field approach. In the second step, combinatorial optimization is employed to identify the best subset of viewpoints from the candidate viewpoints set. This two-step process is then iterated several times to minimize the number of resultant viewpoints and to maximize the coverage ratio. The main contributions of this paper are:

- a novel iterative sampling-based method to generate viewpoints for the "generate-and-test" framework [4];
- a novel viewpoint positions generation method that generates the sampling space by taking the difference of two binary dilation and samples the viewpoint positions within it; and
- a novel potential-field approach with probabilistic distribution to generate viewing directions for candidate viewpoints.

The rest of the paper is organized as follows: The generalized viewpoint planning problem is formulated in Section II,

[1]Wei Jing and Kenji Shimada are with Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, 15213, USA. wj@andrew.cmu.edu, shimada@cmu.edu

[2]Joseph Polden and Wei Lin are with Mechatronics Group, Singapore Institute of Manufacturing Technology, 71 Nanyang Drive, 638075 Singapore; and SIMTech-NUS Joint Laboratory on Industrial Robotics, c/o Department of Mechanical Engineering, National University of Singapore, 117575 Singapore. poldenjw@simtech.a-star.edu.sg, wlin@simtech.a-star.edu.sg

978-1-5090-3762-9/16/$31.00 ©2016 IEEE

followed by details of our proposed method in Section III. In Section IV, computational tests are performed, with the results and discussion. In Section V, a real-world field test of our algorithm with a UAV is performed to demonstrate the applicability of our proposed approach. Finally, conclusions are made in Section VI.

II. Problem Formulation

In visual inspection and shape reconstruction tasks, a UAV navigates about the target object and acquires surface information with an on-board camera (as shown in Figure. 1). The goal of the view planning task is to minimize the number of individual viewpoints while completely covering the surface of the target object. In our application, the camera is attached to the UAV via a gimbal, which gives the camera six degrees-of-freedom (DOF) - three positional DOF and one rotational DOF from the UAV, and another two rotational DOF from the gimbal. The visible region of the camera sensor is defined by two parameters: Field of View (FOV) and Field of Depth (FOD). Since this is a model-based view planning problem, we assume that the rough shapes of the target objects are known, and that they are represented as triangular surface patches.

The view planning problem discussed in this paper can be formulated as a constrained optimization problem:

$$\begin{aligned} &\min \sum x_i, \quad x_i \in \{0,1\} \\ \text{s.t.} \quad & S \subseteq (\cup f(x_i)) \quad \text{for all } i, \end{aligned} \tag{1}$$

where S is the surface patch set of the target object. x_i is the i^{th} viewpoint; $x_i = 1$ means this viewpoint is selected and $x_i = 0$ means it is not selected. f is the function that maps the viewpoint set to the surface set S. The physical interpretation of $f(x_i)$ is the visible surface area from a given viewpoint (x_i). Given this formulation, the objective is to minimize the number of viewpoints required to cover all the surface patches which make up the target object.

III. Proposed Methodology

In this paper, a novel sampling-based view planning method is presented. The proposed approach utilizes an iterative two-step optimization framework, outlined in Figure 2. In the first step, candidate viewpoints are sampled within a constrained sampling space around the target object. These candidate viewpoints are passed to the second step, which selects the best subset of these viewpoints via combinatorial optimization. This two-step process is then iterated several times to improve the coverage ratio and reduce the number of required viewpoints. The remainder of this section presents the preprocessing step and the two iterative steps in more detail.

A. Preprocessing: Subdivision of Target Surface

In order to formulate the combinatorial optimization problem in later section, we firstly preprocess the target object by subdividing the surface of the target object into a mesh of triangular surface patches. In this paper, we use the Bubble Mesh [11] method to generate a well-shaped triangular mesh that preserves the original shape well and also maintains patch uniformity. The subdivision example is shown in Figure. 3.

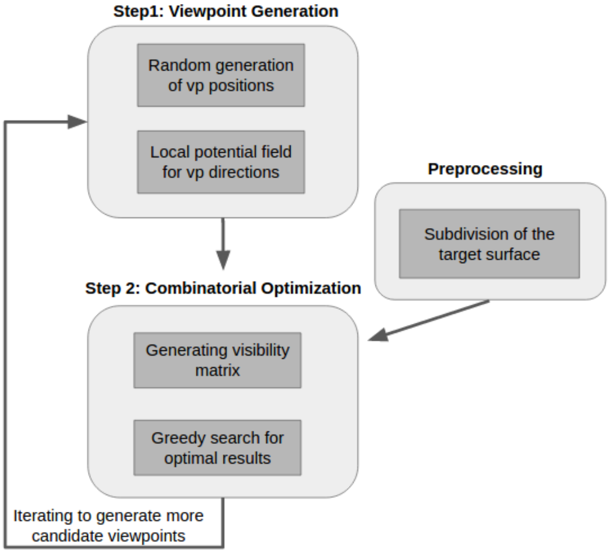
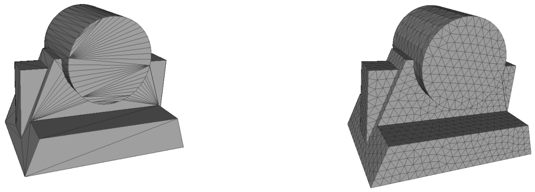

Step1: Viewpoint Generation
Random generation of vp positions
Local potential field for vp directions
Preprocessing
Subdivision of the target surface
Step 2: Combinatorial Optimization
Generating visibility matrix
Greedy search for optimal results
Iterating to generate more candidate viewpoints

Fig. 2: Proposed iterative two-step method for view planning problem.



(a) Original mesh

(b) After subdivision using Bubble Mesh [11]

Fig. 3: Subdivision of model into a set of triangular surface patches

B. Viewpoint Generation

In this work, candidate viewpoints generation involves the following two steps:

- Randomized generation of viewpoint positions in the sampling space generated by taking difference of two dilation.
- Specification of viewpoint directions via probabilistic potential-field method.

1) Random Sampling for Viewpoint Position: Sampling space is required before we can begin sampling candidate viewpoints. In this paper, we obtain the sampling space by using a binary voxel dilation [12] in 3D space. The dilation is a morphology operation defined by:

$$A \oplus B = \{a + b | a \in A, b \in B\}, \tag{2}$$

where A is the original 3D voxel model that is the target object in this paper, and B represents the shape to be dilated.

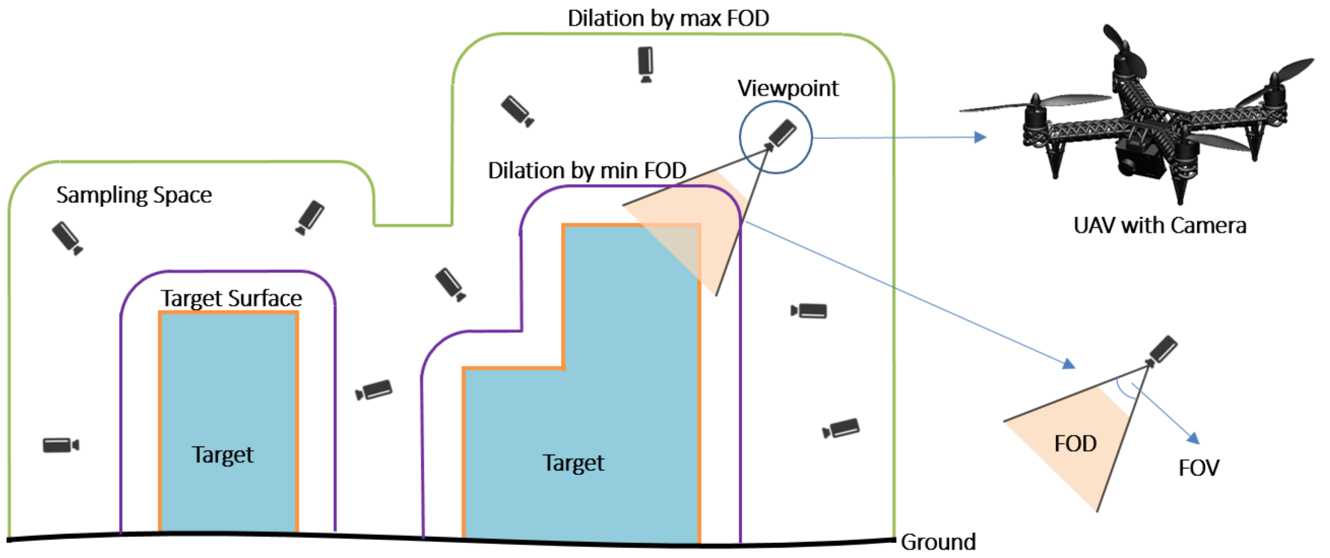


Fig. 1: UAV vision application example: the camera equipped UAV is able to acquire surface information of the target. The camera has a visible region that is defined by the FOV and FOD. The white space between the dilated boundary of maximum FOD (in green), the dilated boundary of minimum FOD (in purple) and ground (in black) is the sampling space in which the candidate viewpoints are sampled (details are presented in Section III.)

As the camera's visibility is constrained by the FOD, two dilations of the target object are performed: First by the camera's maximum FOD, and then by its minimum FOD. The sampling space is obtained by taking the difference between the two dilations. The resultant sampling space can be considered as the complete feasible space, meaning that any viewpoint sampled outside this volume is unable to generate a valid view of any surface patches on the target object. Once complete, the set of candidate viewpoint positions are then randomly sampled inside this volume. The procedure to generate viewpoints is shown in Algorithm 1, noting that the solid model can be found from the mesh model of target object. Once the requisite number of viewpoint positions is generated, we move to assigning a suitable viewing direction for each sample, which is described in detail in Section III-B.2.

Algorithm 1 Randomly generating viewpoint positions with voxel dilation

```
Input: The solid model of target object, $T$; a sphere with
    radius equals to maximum FOD, $S_{max}$; a sphere with
    radius equals to minimum FOD, $S_{min}$; and the number
    of viewpoints to be drawn, $N_{vp}$
Output: The candidate viewpoint set, $V$;
$D_1 \leftarrow$ dilate$(T, S_{max})$
$D_2 \leftarrow$ dilate$(T, S_{min})$
$D \leftarrow D_1 - D_2$
while sizeof$(V) < N_{vp}$ do
    $v \leftarrow$ RandomSampleVP$(D)$
    $V \leftarrow$ append$(V, v)$
end while
return $V$
```

2) Viewpoint Direction Generation: After sampling the candidate viewpoint positions, the next step is to generate their corresponding viewing directions. In this paper, we propose a novel probabilistic potential-field method to generate viewing directions. This is done by considering that nearby surface patches on the target object have an attraction force which is proportional to $1/d^2$. As shown in Eq. (3) below, the viewing direction for each point is calculated by summing nearby surface patch attraction forces. The resultant 3D force vector is then normalized to give a unit vector that specifies viewing direction of corresponding viewpoints.

Formally, the viewing direction $\mathbf{v}$ of a given viewpoint located at $\mathbf{p_{vp}}$ is:

$$\mathbf{v} = \frac{\sum_i^N \frac{K\mathbf{d_i}}{\|\mathbf{d_i}\|^3}}{\|\sum_i^N \frac{K\mathbf{d_i}}{\|\mathbf{d_i}\|^3}\|}, \tag{3}$$

$$\begin{aligned} \text{where:} \quad & \mathbf{d_i} = \mathbf{p_{vp}} - \mathbf{p_{patch_i}} \\ \text{for all:} \quad & \{\mathbf{p_{patch_i}} | (\mathbf{p_{vp}} - \mathbf{p_{patch_i}})^T(\mathbf{p_{vp}} - \mathbf{p_{patch_i}}) \\ & \qquad < d_{max}\}, \end{aligned}$$

where K is a constant value; $\mathbf{p_{patch_i}}$ is the position of i^{th} surface patch; d_{max} defines the maximum inclusion distance, only surface patches located within this distance to the viewpoint will be included in the calculation of mean viewing direction.

After obtaining the viewing direction by potential-field method, we also add variance to the viewing direction. The variance modeled by Gaussian distribution in 3D space is added to the viewing direction, as shown in Eq. (4). The camera on the viewpoint is assuming always pointing to the viewing direction with y-axis pointing upwards.

$$\mathbf{v}' = \frac{\mathbf{v} + \mathbf{x_v}}{\|(\mathbf{v} + \mathbf{x_v})\|}, \tag{4}$$

where $\mathbf{x_v} \sim \mathcal{N}_3(\mathbf{0}, \mathbf{\Sigma})$ is the multivariate Gaussian distribution.

C. Combinatorial Optimization

1) Visibility Matrix: The visibility matrix is a $M \times N$ binary matrix that measures Boolean visibility information of a given surface patch to a particular viewpoint. M represents the number of viewpoints, and N the number of surface patches. A surface patch is visible to a viewpoint if the following conditions are satisfied:

- The surface patch must be in the Field of View (FOV) of the sensor from the viewpoint.
- The surface patch must be in the Field of Depth (FOD) of the sensor from the viewpoint.
- The viewing angle β must be within a certain range predicted by the sensor specifications.
- There must be no occlusion, which means no solid element lies in between the surface patch and the viewpoint. This condition is calculated using a modified version of ray-triangle intersection algorithm [13].

In this paper, we use a pinhole camera model and projection matrix [14] to model FOV visibility. By setting the skew parameter to 0, the ideal intrinsic matrix of a pinhole camera, $\mathbf{K}$, is:

$$\mathbf{K} = \begin{bmatrix} \alpha_x & 0 & p_x/2 \\ 0 & \alpha_y & p_y/2 \\ 0 & 0 & 1 \end{bmatrix} \in R^{3\times3},$$

where α_x, α_y are the focal length in terms of pixel dimensions on x and y axe respectively; p_x, p_y are the number of pixels on the x and y axes, respectively.

The camera's extrinsic matrix, $\mathbf{E}$, is generated from the position, $\mathbf{T}$, and orientation, $\mathbf{R}$, of a given viewpoint:

$$\mathbf{E} = [\mathbf{R}, \mathbf{T}] \in R^{3\times4}.$$

Given both the intrinsic and extrinsic matrix of the camera, the projection of a point $\mathbf{p} = [x, y, z, 1]^T \in R^{4\times1}$ on the image plane can then be calculated as:

$$\mathbf{u} = \mathbf{K} \cdot \mathbf{E} \cdot \mathbf{p}, \tag{5}$$

where $\mathbf{u} \in R^3$.

To ensure visibility, the projection of a given point must be within the pixel range of the camera. Therefore, for a given point, if $0 \leq u_1/u_3 \leq p_x$ and $0 \leq u_2/u_3 \leq p_y$ after projection, then it is considered to be within the visible region of the given viewpoint, otherwise it is considered to be outside the visible region. The overall procedure to generate visibility matrix is shown in Algorithm 2.

Algorithm 2 Generating Visibility Matrix

```
Input: The set of surface patches, $P$; the set of candidate
    viewpoints, $V$; and the sensor parameters, $s$;
Output: The visibility matrix, $A_{vm}$;
for each $v_i \in V$ do
    for each $p_j \in P$ do
        if isVisible($p_j$, $v_i$, $s$) then
            $A_{vm}(i, j)$ = 1
        else
            $A_{vm}(i, j)$ = 0
        end if
    end for
end for
return $A_{vm}$
```

2) Set Covering Problem: With the visibility matrix computed, the view planning problem in this paper can now be formulated as two types of Set Covering Problem (SCP): single coverage problem for visual inspection and multiple coverage problem for shape reconstruction.

The SCP is formulated to find the least number of viewpoints that cover for every surface patch:

$$\begin{aligned} &\min \sum_{i=1}^{n} x_i, \quad \text{where } x_i \in \{0, 1\} \\ &\text{s.t.} \quad A_{vm} \cdot \mathbf{x} \geq \gamma, \end{aligned} \tag{6}$$

where x_i is the i^{th} viewpoint, $x_i = 1$ means this viewpoint is selected and $x_i = 0$ means it is not selected; A_{vm} is the $m \times n$ visibility Matrix; $A_{vm} \cdot \mathbf{x}$ gives a $m \times 1$ vector, which specifies how many times each surface patch has been covered; γ is the coverage frequency, which in this case, is a $m \times 1$ vector. For inspection tasks, each element in γ is set to 1, whilst for reconstruction tasks, they are set to 2.

SCP is a NP-hard problem with no available solution in polynomial time. However there are a number of approximate methods available, such as greedy algorithm [15], Dynamic Programming [16], and Genetic Algorithm [17], which can be used. In this paper, we use a modified greedy search algorithm to solve SCP due to low computational cost and acceptable results for this application. The greedy algorithm used in this paper is defined in Algorithm 3.

D. Iteration of the Two-Step Process

If there remains surface patches on the target object that have not been covered after solving the SCP, an iterative process is employed to generate additional viewpoints which target these specific patches. The same sampling space is used to randomly generate additional viewpoint positions. When calculating viewing directions, however, only the unseen surface patches and the patches visible from less than certain number of viewpoints in previous iterations are considered for the probabilistic potential-field method. The newly generated viewpoints are added to the previous candidate viewpoint set before performing the combinatorial optimization procedure outlined earlier. The iterative process

Algorithm 3 Greedy Search Algorithm for Set Covering Problem

```
Input: The set of surface patches, $P$; the set of candidate viewpoints, $V$; the visibility matrix, $A_{vm}$; and the coverage frequency, $\gamma$
Output: The coverage ratio, $\delta$; the number of resultant viewpoints $n_{res}$; and the resultant viewpoints set $V_{res}$
$V_{res} \leftarrow \emptyset$
$P_{uncover} \leftarrow$ findUncovered($A_{vm}, \gamma$)
$P \leftarrow$ delete($P$, $P_{uncover}$)
$\delta = 1-$ sizeof($P_{uncover}$) / sizeof($P$)
while $P \neq \emptyset$ do
    $v \leftarrow$ maxCover($V, P, A_{vm}$)
    $V_{res} \leftarrow$ append($, V_{res}, v$)
    $V \leftarrow$ delete($V$, $v$)
    for each $p_j \in P$ do
        if count($p_j, V_{res}, A_{vm}$) $> \gamma$ then
            $P \leftarrow$ delete($P$, $p_j$)
        end if
    end for
end while
$n_{res}$ = sizeof($V_{res}$)
return $\delta, n_{res}, V_{res}$
```

is repeated a number of times to further improve the coverage ratio and overall results.

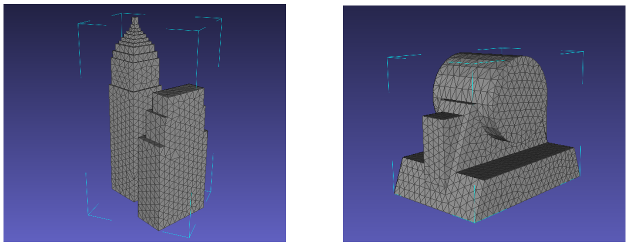
IV. Computational Tests and Discussion

A. Setup

In this section, we conduct a series of computational tests to compare the effectiveness of our method against two existing approaches used in inspection and reconstruction tasks. Our method is compared with two previous methods: offsetting method [7] and viewing sphere method [8][9]. All three methods use the general two-step "generate-and-test" framework, as well as the same optimization strategy in the second step. The difference between these methods is in how the candidate viewpoint positions and directions are generated.

The offsetting method generates candidate viewpoints by directly offsetting a distance from each surface patch. Viewing directions are then generated to point directly towards the corresponding patch. The view-sphere method generates candidate viewpoints on the surface of a virtual sphere enclosing the target object. Viewing directions are generated so that they point towards the centre of the sphere. The same combinatorial optimization strategies are used in all three methods. For the iterative sampling method proposed in this paper, the maximum number of allowable iterations is set to three.

The target objects used for the tests are shown in Figure. 4. The first target object used for view planning computation consists of $2,570$ individual triangular patches and stands at a height of approximately 180 meters. The second target object that consists of $2,566$ triangular patches is a smaller object, approximately $1/100th$ the size of the building structure.



(a) Target object 1 with the bounding box size of $85m \times 100m \times 180m$

(b) Target object 2 with the bounding box size of $1.12m \times 1.37m \times 1.68m$

Fig. 4: The target objects

Two computational tests have been conducted in this paper. Firstly, an inspection task on Target object 1. For this inspection task, surface patches on the target object only need to be covered at least once. The view planning parameters for this test are given in Table I. The second test involves a reconstruction task of Target object 2. For this reconstruction task, surface patches must be covered at least twice for triangulation purposes. The relevant parameters for this view planning process are listed in the table II. In each of the two tests, the three view planning algorithms will be implemented and their performance compared. The results of the tests are presented in the Section IV-B.

TABLE I: View planning parameters for the target object 1

Focal Length (35mm equivalent)	20mm
Camera Pixels	$3,000 \times 2,000$
Viewing Angle	$80°, 90°$
FOD (meters)	(1, 50), (1,100)
Coverage Frequency	Single $\gamma = 1$

TABLE II: View planning parameters for the target object 2

Focal Length (35mm equivalent)	30mm
Camera Pixels	$3,000 \times 2,000$
Viewing Angle	$70°, 80°$
FOD (meters)	(0.5, 5)
Coverage Frequency	Multiple ($\gamma = 2$)

For both tests, the number of viewpoints required n_{res} is calculated by the greedy search method proposed in Algorithm 3. The coverage ratio δ is computed by using the number of covered surface patches divided by the number of total surface patches, which is also from Algorithm 3.

B. Results

1) Test 1: Inspection Application: The results shown in Table III indicate that the proposed method requires fewer individual viewpoints to cover Target object 1 for the inspection application, and it also features the best coverage ratio. On average, our proposed method requires 44.8% and 54.6% fewer viewpoints compared to the other two methods, respectively. The 3D visualization of the resultant viewpoints is presented in Figure 5.

TABLE III: Results from Test 1.

Parameters	Sampling		Offset		SPH	
	n_{res}	δ	n_{res}	δ	n_{res}	δ
$(\mathbf{50}, \mathbf{90}^{\circ})$	37	100%	74	99.9%	82	55.2%
$(\mathbf{50}, \mathbf{80}^{\circ})$	45	100%	92	99.8%	92	54.0%
$(\mathbf{100}, \mathbf{90}^{\circ})$	13	100%	22	100%	46	94.7%
$(\mathbf{100}, \mathbf{80}^{\circ})$	22	100%	35	99.4%	51	89.6%

n_{res} is the number of viewpoints required, and δ is the coverage ratio. $(\mathbf{50}, \mathbf{90}^{\circ})$: means maximum FOD, d_{max} is 50 meters and the maximum viewing angle $\beta_{max} = 90^{\circ}$
"Sampling" refers to the sampling-based method proposed in this paper; "Offset" refers to the offsetting method [7]; "SPH" refers to the sphere method [8][9].

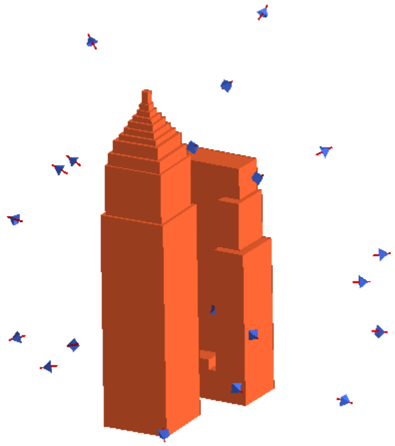


Fig. 5: The 3D visualization of the Target object 1 and planned viewpoints using the proposed method

2) Test 2: Reconstruction Application: The results listed in Table IV show that the proposed method requires fewer viewpoints to cover the Target object 2 for the reconstruction application, whilst maintaining the highest coverage ratio as well. On average, the proposed method requires 44.3% and 9.4% fewer viewpoints compared to the previous two methods on average. The 3D visualization of the resultant viewpoints is presented in Figure 6.

TABLE IV: Results from Test 2.

Parameters	Sampling		Offset		SPH	
	n_{res}	δ	n_{res}	δ	n_{res}	δ
$(\mathbf{5}, \mathbf{80}^{\circ})$	13	100%	25	100%	16	99.8%
$(\mathbf{5}, \mathbf{70}^{\circ})$	19	100%	32	99.4%	19	99.4%

n_{res} is the number of viewpoints required, and δ is the coverage ratio. $(\mathbf{5}, \mathbf{80}^{\circ})$: means maximum FOD, d_{max} is 5 meters and the maximum viewing angle $\beta_{max} = 80^{\circ}$
"Sampling" refers to the sampling-based method proposed in this paper; "Offset" refers to the offsetting method [7]; "SPH" refers to the sphere method [8][9].

C. Discussion

During the computational tests, it is observed that all three methods shares a similar time-frame required to generate their respective solutions. Further inspection of the timing reveals that the computational cost can be broken down into three key parts: The generation of candidate viewpoints, the generation of the visibility matrix and solving the SCP problem. Of these components, the most computationally expensive, by a wide margin, was encountered when generating the visibility matrix. All three methods use the same approach to do this, which takes around 20 minutes to complete (using single thread with Intel Core i5-3320M CPU and 8G RAM laptop). This result is encouraging, as it shows that our proposed method is able to out-perform previously available approaches without any significant penalty in computation time, which further enhances its suitability for real-world applications.

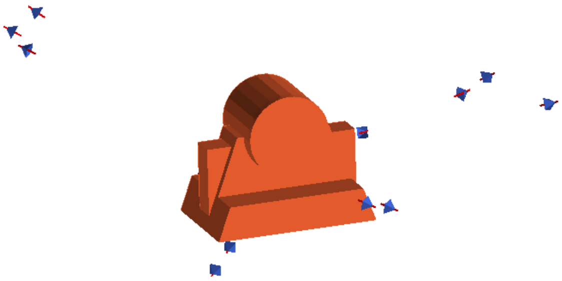


Fig. 6: The 3D visualization of the Target object 2 and planned viewpoints using the proposed method

The results obtained from the tests performed earlier have indicated that our proposed method is able to outperform the others without requiring additional computation time. This can be attributed to a number of factors. Firstly, our method generates viewpoints via randomized sampling, which implies that the method is non-deterministic, allowing performance to be relatively consistent for a variety of different target objects. This is in contrast to the other two methods tested, which generate viewpoints deterministically from a small subset of all feasible space. This can give inconsistent performance with different target geometries, and in some cases, may not be able to provide a solution at all. Additionally, the potential-field method used to generate viewpoint directions ensures the visible surface area of the target object is maximized for each individual viewpoint. This effect is particularly useful for viewing difficult regions of the target object; when the iterative process is initiated, the easier to view surface patches will not be included in the potential-field calculations (as they have already been covered), meaning that all generated viewpoints will then point towards the difficult to capture regions of the target object's surface.

V. FIELD TEST

To demonstrate the effectiveness of the proposed method in a real world reconstruction application, a field test using a DJI phontam 3 UAV has been conducted. The UAV features a camera mounted via a 3-axis gimbal, with a $35mm$ equivalent focal length of $20mm$. The target object is a statue (as shown in Figure 7) with a height of about 3 meters. Since this is a model based view planning exercise, a rough model of the target object is provided in advance in order to conduct the test.

1) Calibrating the Camera: The camera mounted on the UAV is calibrated with a black-and-white chessboard. Eleven pictures of the 9×7 chessboard have been taken at different poses. The resultant calibrated intrinsic matrix $\mathbf{K}$ of the UAV camera is then found to be:

$$K = \begin{bmatrix} 2337.76 & 0 & 1991.83 \\ 0 & 2346.33 & 1466.06 \\ 0 & 0 & 1 \end{bmatrix}$$

The remaining camera parameters for the UAV camera used for view planning are listed in Table. V.

TABLE V: View Planning Parameters for UAV Field Test

Viewing angle	70°
FOD (meters)	(0.5, 5)
Coverage frequency	Multiple ($\gamma = 2$)
Minimum flying height	$0.5m$

2) 3D Reconstruction: The sampling-based view planning algorithm presented in this paper is applied to find the viewpoints required for 3D reconstruction of the statue. The result publishes 16 individual viewpoints required to cover the statue. The UAV is piloted to each of the specified viewpoints to capture an image of the statue. The resulting reconstructed model of the statue, shown in Figure. 8 and Figure. 9, is generated with the software VisualSFM [18][19] and CMP-MVS [20].

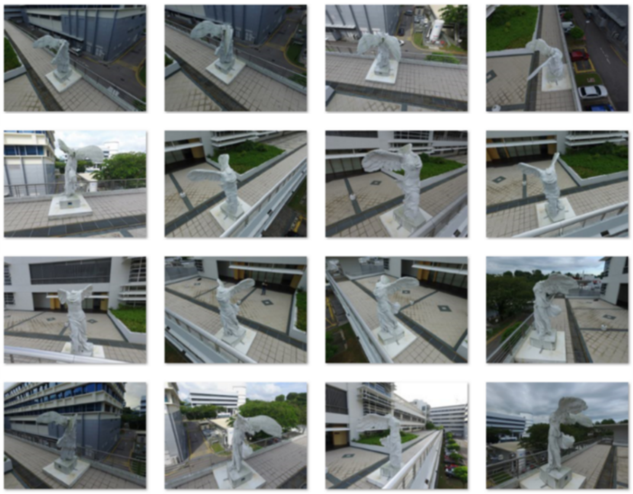


Fig. 7: The picture captured at the planned viewpoints

VI. CONCLUSIONS

In this paper, a novel iterative sampling-based view planning method has been proposed. The proposed approach makes use of an iterative two-step framework where viewpoints are generated through random sampling (to generate viewpoint positions) combined with probabilistic potential-fields (to generate viewpoint directions), and combinatorial optimization to select the most suitable viewpoints from those that have been sampled. We show that our method outperforms two existing view planning algorithms. The proposed method has also demonstrated its ability in solving both inspection and reconstruction tasks by generating solutions featuring fewer viewpoints and higher overall coverage ratios. Finally, the method is tested in a real-world field application where a UAV equipped with camera is used to reconstruct a 3D model of an outdoor statue. Future work planned for this approach to viewpoint planning will involve adaptation of this algorithm to non-model based shape reconstruction tasks and combination of the proposed view planning algorithm with path planning algorithms.

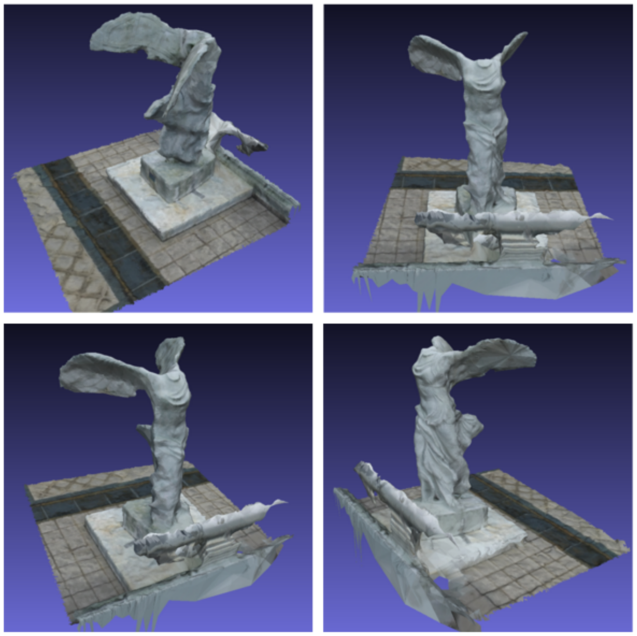


Fig. 8: Different viewing angles of the 3D reconstruction results of the statue

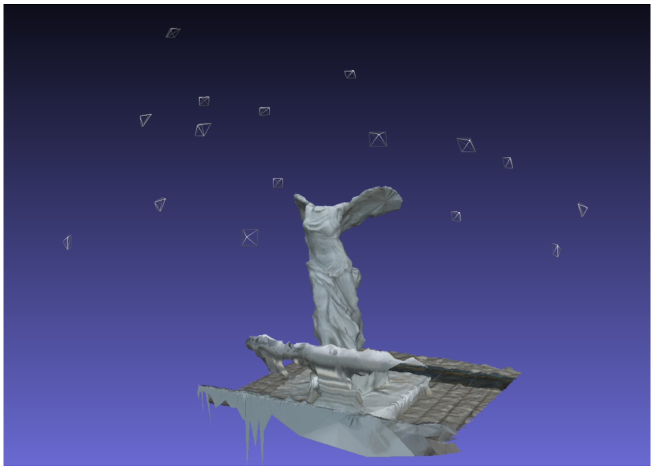


Fig. 9: The 3D reconstruction result of the statue with the camera positions

ACKNOWLEDGMENT

The authors acknowledge the support from the Collaborative Research Project under the SIMTech-NUS Joint Laboratory on Industrial Robotics, Ref: U13-R-024AU and support from the SERC Industrial Robotics Program Work Package 8, project number: 122510008, and the support from A*STAR Graduate Academy of the Agency for Science, Technology and Research (A*STAR), Singapore.

REFERENCES

[1] M.-D. Dios and A. Ollero, "Automatic detection of windows thermal heat losses in buildings using uavs," in *WAC'06. World Automation Congress, 2006.* IEEE, 2006, pp. 1–6.

[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] K. A. Tarabanis, R. Y. Tsai, and P. K. Allen, "The mvp sensor planning system for robotic vision tasks," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 72–85, 1995.

[4] W. Scott, G. Roth, and J.-F. Rivest, "View planning for automated 3d object reconstruction inspection," *ACM Computing Surveys*, vol. 35, no. 1, 2003.

[5] 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.

[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] 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.

[9] 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.

[10] L. Geng, Y. Zhang, J. Wang, J. Y. Fuh, and S. Teo, "Mission planning of autonomous uavs for urban surveillance with evolutionary algorithms," in *IEEE International Conference on Control and Automation*. IEEE, 2013, pp. 828–833.

[11] 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.

[12] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 4, pp. 532–550, 1987.

[13] T. Möller and B. Trumbore, "Fast, minimum storage ray-triangle intersection," *Journal of graphics tools*, vol. 2, no. 1, pp. 21–28, 1997.

[14] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[15] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of operations research*, vol. 4, no. 3, pp. 233–235, 1979.

[16] Q.-S. Hua, Y. Wang, D. Yu, and F. Lau, "Dynamic programming based algorithms for set multicover and multiset multicover problems," *Theoretical Computer Science*, vol. 411, no. 26, pp. 2467–2474, 2010.

[17] W.-C. Huang, C.-Y. Kao, and J.-T. Horng, "A genetic algorithm approach for set covering problems," in *IEEE World Congress on Computational Intelligence*. IEEE, 1994, pp. 569–574.

[18] C. Wu, "VisualSFM: A visual structure from motion system," *URL http://ccwu. me/vsfm/*, 2011.

[19] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *2011 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 3057–3064.

[20] 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.