IMAV2017-40
http://www.imavs.org/pdf/imav.2017.40

3D Reconstruction of Complex Structures with Online Profiling and Adaptive Viewpoint Sampling

Abdullah Abduldayem, Dongming Gan, Lakmal Seneviratne, Tarek Taha*
Khalifa University, Abu Dhabi, United Arab Emirates

ABSTRACT

A modified Next Best View (NBV) approach is presented to improve the 3D reconstruction of complex symmetric structures. Two new stages are introduced to the NBV approach; A profiling stage quickly scans the structure of interest and builds a rough model called the ”profile”. The profile is relatively sparse but captures the major geometric features of the structure. A symmetry detection stage then determines major lines of symmetry in the profile and labels points of interest. If a point exists in known space but its mirror image lies in unknown space, the mirrored point becomes a point of interest.

The reconstruction is performed by a sensor mounted on an Unmanned Aerial Vehicle by using a utility function that incorporates the detected symmetry points. We compare the proposed method with the classical information gain utility function in terms of coverage completeness, total iterations, and travel distance.

1 INTRODUCTION

Once restricted to controlled industrial environments, robots have become increasingly autonomous in the past decade. One task being delegated to robots is the automatic 3D reconstruction of objects and structures where there is a need for high density reconstructions that ensure complete coverage of the structure. This technology has applications in inspection, model scanning and in preservation of archaeological artefacts in digital form. Performing the inspection task manually is time-consuming with no guarantees on reconstruction density, coverage completeness or repeatability.

In some cases, a model or blueprint of the structure is provided which can be used to pre-compute paths for the robot to follow as it reconstructs the structure [1, 2]. However, these models are not available or provided in many cases.

To overcome this problem, the robot iteratively constructs a map as it navigates around the object of interest. One such method is the so-called **Next Best View (NBV)** approach [3, 4, 5] which attempts to determine where to place the sensor next in order to obtain the most new information about the structure.

The basic NBV methodology works as follows: Starting from an initial position, the robot senses its surroundings and creates an initial map of the structure. The robot then generates a set of candidate viewpoints to move to next, evaluates each of these viewpoints, and finally moves to the view that maximizes the information gained about the structure. A utility function is usually used to evaluate these viewpoints by weighing factors such as information gain and effort required to reach that viewpoint. This process is repeated over and over, selecting new viewpoints at each iteration until the structure is fully modelled, or a termination criteria has been met.

2 BACKGROUND

Model-less based approaches incrementally build the map as they explore. Typically, these make use of a Next Best View (NBV) approach as described in the introduction. The method is often broken down into stages, with the two main components being Viewpoint Sampling and Viewpoint Evaluation.

2.1 Viewpoint Sampling

Some techniques that have been explored to generate candidate viewpoints are frontier approaches, Rapidly Exploring Trees (RRT) and discretized state space approaches.

A **frontier** is a boundary between explored and unexplored space [6]. The frontier is determined by constructing an occupancy grid which marks whether a cell is free, occupied or unexplored. Unexplored cells which are adjacent to free cells are frontier cells, and the chain of adjacent frontier cells is a frontier.

Frontier methods perform well indoors and in constrained areas. In these environments, the frontier exists only at the end of a corridor or start of a room. However in an outdoor setting, the sky represents a large unexplored space and, if there are no obstructions, the frontier can potentially reach the horizon and become enormous. Bircher et al. [5] shows that frontier approaches stall in outdoors environments, with most of this time spent on evaluating the multitude of viewpoints.

A related technique for reconstruction is the **boundary search** method [7]. Rather than finding the edge of explored *space*, it locates the edge of the explored *object*. Poses near the boundary are sampled, and the quality of the poses can be improved by estimating the shape of the unseen surface at the boundary [7]. A similar method locates ”barely visible” surfaces whose normals exceed a defined visibility angle [8].

*Email: {abdullah.dayem, dongming.gan, lakmal.seneviratne, tarek.taha}@kustar.ac.ae

Rapidly-exploring Random Trees (RRT) sample viewpoints randomly and connects these viewpoints in a tree-like structure, expanding throughout the explored space. Each branch represents a set of viewpoints to be visited in sequence, and the branch which maps the most unmapped space is selected. These trees can be grown using RRT or the RRT* variant [9].

Rather than executing an entire branch at once, Bircher et al. [5] determines the best branch and executes only its first edge. The remainder of the branch is used to initialize a new tree, preserving the original path while allowing for new paths to grow as the world map is updated. This method prefers to yaw in place to minimize cost penalty while the frontier method always moves to edges of known space. In Bircher's tests, RRT performed similar to frontier in indoor environments but vastly outperforms it outdoors. This is because the number of voxels in the frontier representation grows exponentially with size so it takes longer evaluate all the possible frontiers.

The **discretized state space** —also known as state lattice, configuration space or discretized joint space— takes each degree of freedom (DOF) and divides its range into a number of discrete steps. The possible poses that an n-DOF robot can take are thus visualized as points lying in a discretized n-dimensional space. While this technique is usually applied to manipulators with fixed joint range [10], it can also be used on a mobile robot if an estimated volume for the workspace can be obtained. Those working with UGVs may use 2-D position and yaw while UAVs may use 3-D position and yaw [11]. Additional DOFs may be added if a pan-tilt camera is used to alter the pitch of the camera as well.

With each additional DOF, the number of points in the discretized space increases exponentially. Similarly, large scenarios and fine resolution increase the number of viewpoints drastically, making it unsuitable for large-scale high precision applications. In these cases, a method must be used to limit the selection before evaluating the next best view.

2.2 Utility Functions

A common strategy is to simply move to the closest viewpoint. This has been found to reduce movement time, processing time and effort but it results in reduced information gained per viewpoint [12]. Traditionally the robot would simply move to the closest frontier, but a variety of utility functions have been employed to achieve improved results [3].

A utility function is any function that attempts to rank the sampled viewpoints. Factors included in utility functions may include distance to viewpoint, view angle with respect to surface normal [7], and distance to hazards [13].

3 Proposed Method

Rather than attempting to tackle the NBV problem blindly, we propose a quick profiling stage. Before applying NBV, an unmanned aerial vehicle (UAV) scans the structure to obtain a rough initial model of the structure. The initial model will henceforth be referred to as the **profile**, as it shows the general shape of the structure and captures its significant geometric features. The process of obtaining the profile is called **profiling**.

After the profiling step, the main steps of the standard NBV method are executed as shown in Figure 1. The method samples a finite number of viewpoints, evaluates them and determines the next best viewpoint. The UAV then moves to the selected viewpoint and updates the map and structure representations. The termination condition is evaluated, and if the criteria is not satisfied, the process is repeated.

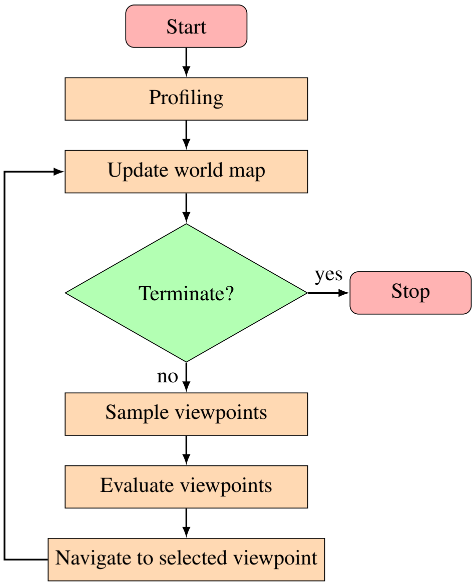

Figure 1: The basic components of the proposed NBV method

3.1 Structure Representation

As new point data is acquired by the sensors, two representations of the structure are updated: a point cloud and an occupancy grid. The occupancy grid divides space into a grid and stores a probability of occupancy in each cell. A probability of 1 means the cell is fully occupied, 0 is completely free and 0.5 is unknown as it is equally likely to be occupied or free. By using the notion of entropy from information theory, the amount of new information remaining in the scene can be computed. The entropy e_i of the cell i with occupancy probability p_i can be found by:

$$e_i = (p_i) \log (p_i) + (1 - pi) \log (1 - pi) \tag{1}$$

The graph of e_i has a peak at p_i = 0.5, meaning unknown cells have the most information while completely free and completely occupied cells have no new information.

The occupancy grid is unsuitable for reconstructing the final structure due to its large spacial resolution. Instead, the final mesh of the structure is generated from a point cloud, which is denser and more precise. Since the point cloud does not differentiate between free and unexplored space, it is necessary to make use of both occupancy grids and point clouds.

3.2 Profiling

The profiling stage initializes both the occupancy grid and the point cloud to guide further exploration.

To capture the profile, we have opted to use laser range data. While it is possible to obtain the profile using visual data, these tend to have poorer distance estimation at longer ranges and require multiple viewpoints to establish spatial positioning of an observed point. Visual sensors also rely on having a feature-rich environment to obtain features, so a reflective or plain surface would be captured with a relatively low point density.

On the other hand, range data has the benefit of collecting reliable 3D points from a single reading. LIDARs can take readings in nearly 360 degrees at distances up to 100 meters. The combined angular and distance capabilities allow the system to gather data about a large structure quickly.

Laser spatial resolution degrades with distance due to increasing distance between the radial rays, so point density will be lower at farther regions of the structure. Some areas may also be occluded, so it is necessary to explore the structure up close with a higher density sensor.

3.2.1 Adaptive Circular Profiling

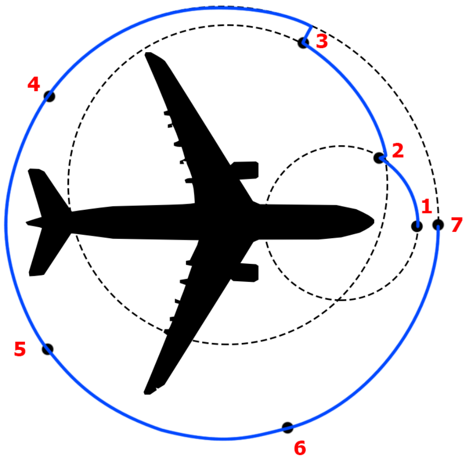

Figure 2: Example of profiling from top view. Dotted lines represent virtual cylinders that encapsulate all point data seen until that time instance. At each of numbered dot, the UAV scans vertically, updates the profile, and recomputes the radius and center of the bounding cylinder. The UAV then travels to the next waypoint and repeats.

We propose an adaptive method of obtaining a profile by moving in a circle around the structure whose radius expands to encapsulate the structure. The vehicle travels around a circle that fully encapsulates the points observed thus far. As the vehicle travels around the circle, the profile is updated and the centroid and bounding radius of the cylinder is adjusted. The procedure is illustrated in Figure 2 and described more formally in Algorithm 1.

Algorithm 1 Profiling: Adaptive Circular

Input:

- N — Number of viewpoints around structure for profiling
- d — Minimum distance for obstacle avoidance

Output:

- r_{occ} — an occupancy grid
- r_{cloud} — a point cloud

```
θ_s ← Angle between the global x axis and the vector from the UAV's starting position to the center of the workspace
$\theta \leftarrow \theta_s$
$\phi \leftarrow 2\pi/N$
while $(\theta - \theta_s) < 2\pi$ do
    Scan by moving vertically between $z_{min}$ and $z_{max}$
    UpdateProfileMaps($r_{occ}$, $r_{cloud}$)
    $proj$ ← ProjectOntoXYPlane($r_{cloud}$)
    $c$ ← ComputeCentroid($proj$)
    $r$ ← GetBoundingRadius($proj$, $c$)
    $r \leftarrow r + d$
    MoveToCircleCircumference($c, r, \theta$)
    $\theta \leftarrow \theta + \phi$
    MoveAlongCircleCircumference($c, r, \theta$)
end while
```

3.2.2 Symmetric Prediction

The profile is likely to contain holes due to occlusion or limited sensor range. Many structures have symmetry, whether it is rotational, translational, intrinsic, extrinsic or otherwise. We can extract additional information about the structure by exploiting this property. In our application, we focus on reflectional symmetry.

In order to detect the line of symmetry, the following approach was taken:

1. Compute keypoints in the profile point cloud using Fast Point Feature Histograms (FPFH) [14]. By focusing on these keypoints, the number of points under consideration is reduced significantly.
2. Compute features for each keypoint using a variant of the Scale Invariant Feature Transform (SIFT) that operates on 3D point clouds. This variant has been implemented by Michael Dixon for the Point Cloud Library [15].

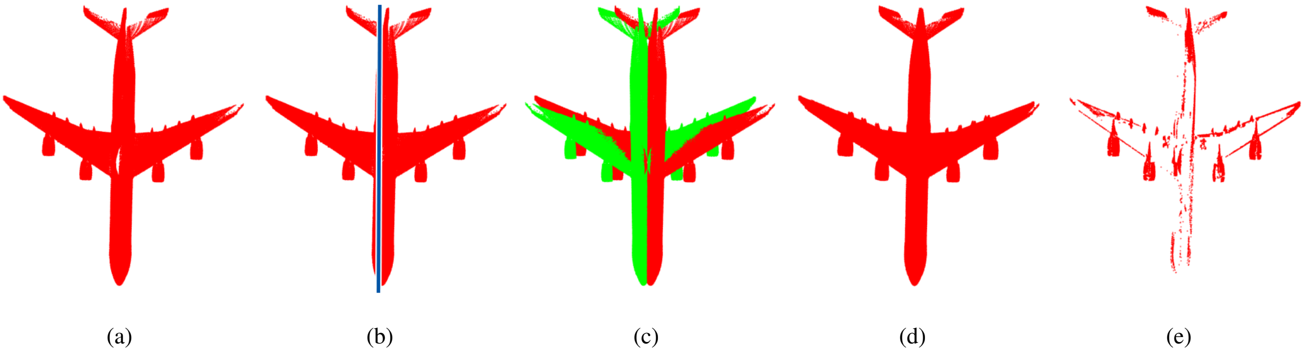

Figure 3: The symmetry-detection process for prediction. **(a)** The profile obtained by the simulated UAV. Some gaps exist in the model. **(b)** The detected line of symmetry shown in blue. **(c)** The profile is mirrored across the line of symmetry and shown in green. **(d)** The mirrored image is aligned with the original using ICP and merged to create the final prediction. **(e)** The additional points obtained from the symmetry process. These are visualized by subtracting the point cloud in (d) from (a)

3. Match each keypoint with the keypoint that has the most similar features (ie. Euclidean distance between features is minimum)
4. Fit a plane between each pair of points
5. Determine the plane of symmetry by performing mean shift clustering on all plane parameters computed in the previous step. Seen in Figure 3(b).
6. Create a copy of the profile point cloud and reflect it across the line of symmetry, as shown in Figure 3(c). This copy is called the **prediction**.
7. The predicted point cloud can be corrected by applying Iterative Closest Point (ICP) [16] between it and the profile, as shown in Figure 3(d).

Once we have obtained the predicted symmetric point cloud, we use it to modify the occupancy grid. Cells that are unknown but have a predicted point within them are updated to an occupancy value of 0.6. This update indicates that we have some confidence that the cell is occupied, but not as confident as a cell that has been directly observed.

3.3 Viewpoint Sampling

As there are an infinite number of poses to consider, attempting to determine the absolute best possible position in continuous space is intractable [12]. Instead, only a few selected viewpoints are sampled. To generate those viewpoints, we use a technique called **constant grid viewpoint sampling (CGVS)**.

The CGVS approach samples a few points in the vehicle's nearby vicinity in both linear and anglar space, as shown in Figure 4. While this method is not globally optimal, it ensures viewpoints in the nearby vicinity are sampled first to reduce travel cost.

This method can be generalized so that the sampling distance can vary. This may be done by multiplying linear distances with a scale factor of σ^α, where $\sigma > 1$ and $\alpha \geq 0$. This scaling approach extends CGVS and can be performed dynamically, giving rise to **adaptive grid viewpoint sampling (AGVS)**. The set of sampled viewpoints is given by Equation 2 with respect to the vehicle's local frame.

If the value of σ is fixed, the value of α must vary in order to change the scaling. Scaling occurs when the method fails to obtain sufficient entropy reduction after a given number of iterations. The criteria for "sufficient entropy reduction" is explained in Section 3.5. If Equation 6 is satisfied for $N = 3$ and $\Delta E_{v,threshold} = 0.005$, then α is incremented by one, otherwise it is reset to zero.

Finally, a viewpoint is considered **invalid and discarded** if it touches an occupied or unknown cell, collides with the structure or lies outside the workspace limits. If the method is unable to generate any valid viewpoints, the NBV process terminates.

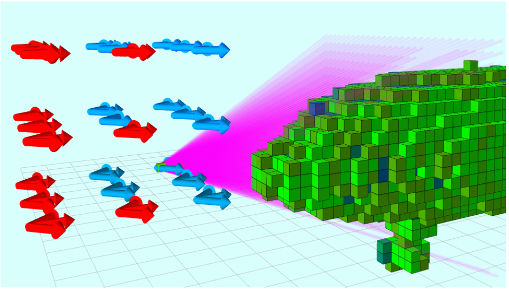

Figure 4: Constant grid viewpoint sampling method. Blue arrows represent valid viewpoints while red ones are rejected due to collision with the structure. Magenta lines represent the sensor's viewpoint at the selected viewpoint. Green voxels represent occupied space, while blue voxels show unknown space.

$$V_{CGVS} = \left\{ \begin{bmatrix} i \times \Delta d \times \sigma^{\alpha} \\ j \times \Delta d \times \sigma^{\alpha} \\ k \times \Delta d \times \sigma^{\alpha} \\ m \times \Delta\theta \end{bmatrix} \middle| \begin{matrix} i,j,k,m \in \{-1,0,1\}, \\ \{i,j,k,m\} \neq \{0,0,0,0\} \end{matrix} \right\} \tag{2}$$

The first three elements of V_{CGVS} represent the spacial increment (x,y,z), while the last element represents the angular increment (yaw). The values of Δd and $\Delta\theta$ are in meters and radians, respectively. In our experiments, these values are set to $\Delta d = 1.0$m and $\Delta\theta = \pi/4$rad as that creates some overlap between viewpoints while providing the opportunity to discover new information. The value of σ is a fixed scale factor, while α is the number of times to apply the scale. We have used values of $\sigma = 1.5$ and $\alpha \in [0, 5]$ so the total scale factor varies from 1 to 7.594.

3.4 Viewpoint Evaluation

To determine which of the sampled viewpoints is the next best view, we select the viewpoint that maximizes a utility function. The utility function operates on a section of the point cloud and/or occupancy grid that is visible at that particular viewpoint.

3.4.1 Utility Function

The Information Gain (IG) metric is often used in literature [5, 3, 17], and will thus be used as a benchmark. It measures the total entropy (see Equation 1) of cells in a given viewpoint.

To determine which cells are visible from a given viewpoint, rays are cast according to the sensor's field of view. The ray passes through free or unknown cells, and stops once it collides with the first occupied cell in its path. The entropy of the cells along and at the endpoint of the ray trajectory is computed as in Equation 3

The total information H_k in a given viewpoint k which is able to see the set of cells V is given by:

$$H_k = \sum_{i \in V} e_i \tag{3}$$

In the proposed method, we make use of Equation 3 using our modified occupancy grid with predicted voxels.

3.4.2 Constraints

In the case where profiling is not performed, there is a large amount of unknown information surrounding the structure. Left on its own, the vehicle would prefer to select viewpoints with purely unknown cells and would move away from the structure. To prevent this, two restrictions are implemented.

First, viewpoints that do not observe at least 1 occupied cell are rejected. This encourages the vehicle to continue to observe the structure and overlap with previous observations.

Second, the structure lies within a bounded workspace. Any measurement outside those bounds is regarded as having zero utility so that even if the first restriction is removed, the vehicle would stay within the given bounds.

3.5 Termination

No consensus has been reached for a suitable termination condition as it varies between applications, and is also difficult to quantify the desired criteria. Ideally, the process should terminate once the system has achieved coverage completeness, but it is difficult to compute coverage without the original model for reference. The work in [18] terminates if the percentage difference in total entropy reduction has fallen below a threshold, while [12] terminates if no view has an IG above a certain threshold.

Both these methods have variable results. If the total number of unknown cells is very large, percentage difference in entropy reduction will be very small. This can happen if the size of the workspace increases, the occupancy cell size is reduced or many unknown spaces remain after the profiling stage. Similarly, it is possible to be temporarily trapped in local minima, so it may not be fair to terminate if IG reduction momentarily stagnates.

3.5.1 Entropy Change Per Viewable Voxel

To tackle the shortcomings of the previous termination conditions, we propose a termination condition based on the entropy reduction normalized by the maximum number of cells that can be observed from a given view. This gives average entropy reduction per cell and is given in Equations 4 and 5:

$$\Delta E_v = \frac{E_{n-1} - E_n}{C_{view}} \tag{4}$$

$$C_{view} = V_{frustum}/V_{cell} \tag{5}$$

where ΔE_v is the entropy change per cell, E_n is the total entropy at iteration n, C_{view} is the number of viewable cells, V_{cell} is the volume of a cell, $V_{frustum}$ is the volume of the view frustum which accounts for the camera's field of view.

In addition to counting the number of complete cells within a viewpoint, the value of C_{view} also includes partial voxels at the boundaries of the frustum. Sensor readings and raytracing can pass through these cells on the edge of the frustum, so it is necessary to count them as well.

We consider the NBV process has terminated if the condition in Equation 6 is true for more than N consecutive iterations. The value of N is selected so that the method does not terminate prematurely if it is stuck in a local minimum. The magnitude of ΔE_v is used rather than the raw value to account for loss of information. Entropy may increase if an obstacle is introduced or removed, or during raytracing with large occupancy cells from a variety of angles.

$$|\Delta E_v| < \Delta E_{v,threshold} \tag{6}$$

By computing the average entropy change in a single voxel, it is possible to have a single threshold (or tight range of thresholds) across a variety of scenario sizes and occupancy cell resolutions.

4 Experiments

4.1 Environment

Simulation experiments were performed on an Alienware-X51-R2 desktop (Intel Core i7-4790 @ 8 x 3.60 GHz, 15.6 GB RAM, no GPU acceleration). The NBV framework was implemented on Ubuntu 14.04 using the Robot Operating System (ROS-indigo) and simulated with Gazebo. The occupancy grid was represented with the OctoMap library [19] while point cloud data was processed using the Point Cloud Library (PCL) [15].

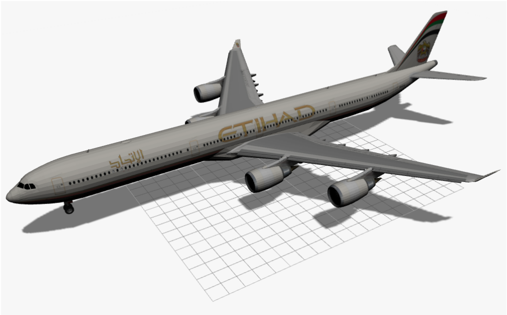

Figure 5: The aircraft used as the object of interest, chosen for its complex geometry

An aircraft was used as the object of interest, as shown in Figure 5 (Dimensions: $35 \times 30 \times 7$ m^3). The aircraft's complex geometry makes it an interesting case study; it has numerous curved surfaces, overhangs, occlusions and is large enough to demonstrate the scale of the problem being tackled. The aircraft also exhibits numerous symmetries that can be exploited.

The vehicle used is a UAV with two sensors, namely a laser scanner and RGB-D camera with specifications as in Table 1. The laser scanner covers a wide angle with a large range, making it suitable for obtaining many points during the large sweeping motions of the profiling stage. The laser scanner is mounted with a slight downwards pitch to ensure it can capture any skyward-facing surfaces. The RGB-D camera is used to obtain dense points at a closer range to create the highly detailed final reconstruction.

A circular collision box is constructed around the UAV to check for collisions with the structure, and its motion is constrained within a workspace of size $40 \times 40 \times 10$ m^3.

4.2 Procedure

Three main tests are performed; first, the classical NBV approach is performed without any profiling or predictive stage (Scenario 1). Second, profiling is performed, but no symmetry prediction is employed. The NBV approach is run using the profile as the initial map (Scenarios 2 & 3). Finally, both profiling and symmetry are performed before starting NBV (Scenarios 4 & 5).

Specification	Laser Scanner	RGB-D camera
Horizontal FOV	180°	60°
Vertical FOV	N/A (single layer)	45°
Resolution	0.25°	640×480 px
Range	0.1–30 m	0.5–8.0 m
Mounting orientation (Roll, Pitch, Yaw)	(0°, 10°, 0°)	(0°, 0°, 0°)

Table 1: Sensor specifications

To further enhance the coverage completeness, the CGVS and AGVS approaches are compared. In this context, CGVS means the value of α is fixed to zero in Equation 2. The value of α increments by one whenever the condition in Equation 6 is met for $N = 3$ and $\Delta E_{v,threshold} = 0.005$. Similarly, the termination condition is met if the Equation 6 is satisfied for $N = 5$ and $\Delta E_{v,threshold} = 0.001$.

For each scenario, the sensor begins in 5 predefined starting locations and the entire process runs until termination. The result of the 5 runs is averaged to give an indication of the average performance for each scenario.

5 Results

The final reconstructions are shown in Figure 6 and the results are plotted in Table 2, where the distance travelled, entropy reduction and coverage are compared. Coverage is evaluated by initializing two new occupancy grids: one constructed from the final point data and another from points sampled from the original mesh. The percentage of matching cells between the two grids is used to determine the coverage.

By changing the size of cells in the two grid, it is possible to measure two different types of coverage. Performing the matching with large voxels (eg. 0.5m) determines how much of the **overall shape** has been captured. On the other hand, matching smaller voxels (eg. 0.05m) gives an indication of how **detailed** the final model is.

Table 2 shows that compared to the full NBV approach (Scenario 1), the profile alone is able to capture the general shape of the structure (86.7% vs 86.9%) with lower travelling cost. However, since the profiling is performed from far away, the resolution is low and hence the density of the reconstructed structure is low (15.1% vs 73.3%).

When the NBV method uses the profile, we see that mid- and large-scale overage improve despite travelling roughly half the distance. While dense coverage did fall, it is mostly due to the lower number of iterations and distance travelled.

The dense coverage (resolution of $0.05m$) of the adaptive method consistently exceeds that of the constant grid method. This is mainly due to the vehicle being able to escape lo-

Scenario	Profiling	Symmetry Prediction	Viewpoint Sampling	Iterations	Distance (m)	Total Entropy Reduction	Coverage (Res=0.05m)	Coverage (Res=0.10m)	Coverage (Res=0.50m)
Profile	—	—	—	—	320.0	927,405	15.1%	61.6%	86.7%
1	No	No	CGVS	859	1,039.9	289,439	73.3%	78.7%	86.9%
2	Yes	No	CGVS	301	538.7	929,483	62.7%	89.0%	89.7%
3	Yes	No	AGVS	301	714.5	**929,509**	71.8%	88.0%	97.7%
4	Yes	Yes	CGVS	217	**526.6**	928,323	61.5%	87.7%	97.4%
5	Yes	Yes	AGVS	257	665.2	928,744	**86.5%**	**94.9%**	**99.5%**

Table 2: Results of the experiment

cal minima by being able to take larger steps. However, this caused travel distance to increase by approximately 30%.

Looking at entropy reduction, we can see that the profile alone is more effective at reducing entropy than the classical NBV method (Scenario 1). Since the LIDAR has long range and most of the space around the structure is empty, it converts most of the unknown cells to free cells. There are additional entropy reductions in Scenarios 2–5, but the value is much smaller as the methods are mainly observing occupied cells with a few scattered unknown cells.

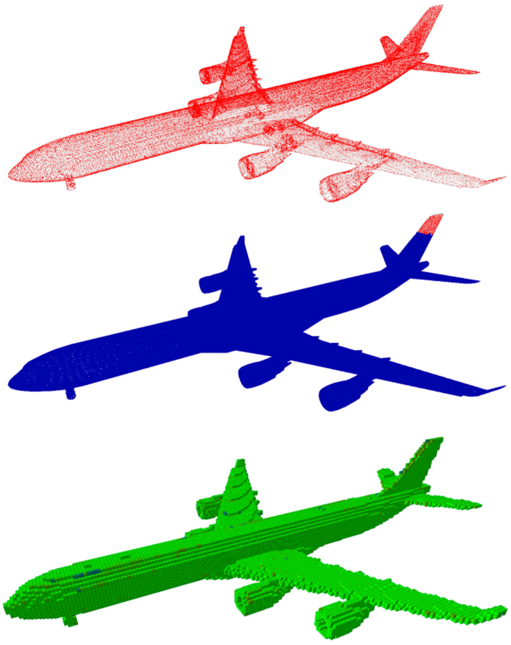

Figure 6: Final representation of the structure. **Top:** The profiled point cloud, representing the partial model obtained at the start of the process. **Center:** The final point cloud after NBV. Note the tail has not be visited during NBV¿ **Bottom:** The final occupancy grid, where green represents an occupied cell and blue represents unknown.

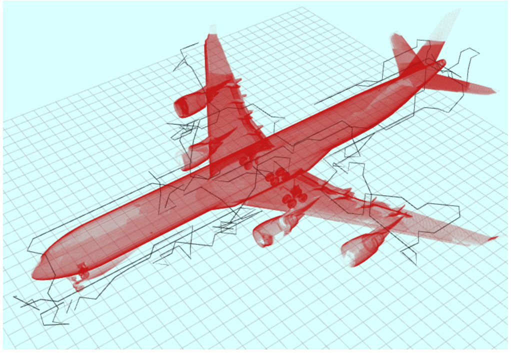

Figure 7: The trajectory taken by the drone during Scenario 4

6 Conclusions and Future Work

We have proposed a method of improving the Next Best View approach for the reconstruction of large, complex structures. A rough scan of the structure is performed to obtain its profile, which is then checked for mirror symmetry. The symmetry is used to make predictions about the shape of the structure to guide the NBV approach. We have also proposed a termination condition based on the average entropy change per cell, and an adaptive viewpoint sampling method based on the same principle.

The introduction of the profiling step shows improvement in coverage completeness with slightly reduced distance. The symmetry prediction performs similar to the classic method and occasionally outperforms it. Finally, the adaptive sampling method is able to escape local minima and brings the coverage completeness closer to 100% while given relatively dense reconstruction.

To further improve this method, we plan to develop a more holistic utility function by incorporating distance and/or density measures. Including the distance may help reduce the total distance travelled while targeting areas with low point density to improve the reconstruction density.

Another possible improvement is the inclusion of frontier methods. Be focusing directly on frontiers and creating viewpoints towards them, it may be possible to obtain a more globally optimal solution.

References

[1] Andreas Bircher, Mina Kamel, Kostas Alexis, Michael Burri, Philipp Oettershagen, Sammy Omari, Thomas Mantel, and Roland Siegwart. Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots. *Autonomous Robots*, pages 1–20, 2015.

[2] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, and G. Cai. Gpu accelerated coverage path planning optimized for accuracy in robotic inspection applications. In *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4, Oct 2016.

[3] Dirk Holz, Nicola Basilico, Francesco Amigoni, and Sven Behnke. Evaluating the efficiency of frontier-based exploration strategies. *Isr/Robotik 2010*, 2010.

[4] Simon Kriegel, Tim Bodenmüller, Michael Suppa, and Gerd Hirzinger. A surface-based next-best-view approach for automated 3d model completion of unknown objects. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4869–4874. IEEE, 2011.

[5] 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*, 2016. (accepted).

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

[7] Simon Kriegel, Christian Rink, Tim Bodenmüller, and Michael Suppa. Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, 10(4):611–631, 2015.

[8] Souhaiel Khalfaoui, Antoine Aigueperse, Yohan Fougerolle, Ralph Seulin, and David Fofi. Online fully automated three-dimensional surface reconstruction of unknown objects. In *The International Conference on Quality Control by Artificial Vision 2015*, pages 95340Z–95340Z. International Society for Optics and Photonics, 2015.

[9] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

[10] Gavin Paul, Phillip Quin, Andrew Wing Keung To, and Dikai Liu. A sliding window approach to exploration for 3d map building using a biologically inspired bridge inspection robot. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on*, pages 1097–1102. IEEE, 2015.

[11] Lionel Heng, Alkis Gotovos, Andreas Krause, and Marc Pollefeys. Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1071–1078. IEEE, 2015.

[12] Phillip Quin, Gay Paul, Alen Alempijevic, Dikai Liu, and Gamini Dissanayake. Efficient neighbourhood-based information gain approach for exploration of complex 3d environments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1343–1348. IEEE, 2013.

[13] Souichiro Oshima, Shingo Nagakura, Jeong Yongjin, Akihiro Kawamura, Yumi Iwashita, and Ryo Kurazume. Automatic planning of laser measurements for a large-scale environment using cps-slam system. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4437–4444. IEEE, 2015.

[14] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[15] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[16] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[17] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. 2016.

[18] Gavin Paul, Stephen Webb, Dikai Liu, and Gamini Dissanayake. Autonomous robot manipulator-based exploration and mapping system for bridge maintenance. *Robotics and Autonomous Systems*, 59(7):543–554, 2011.

[19] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.