Coverage-Based Next Best View Selection

Samuel Cunningham-Nelson[1,2], Peyman Moghadam[1*], Jonathan Roberts[2] and Alberto Elfes[1]
[1]Autonomous Systems, CSIRO, Australia
[2]Queensland University of Technology, Australia
sam.cunningham-nelson@csiro.au, peyman.moghadam@csiro.au,
jonathan.roberts@qut.edu.au, alberto.elfes@csiro.au

Abstract

Generating 3D scans of environments and objects is an invaluable tool, which can be used in many industries. A small handheld device known as *HeatWave* can create a 3D environment map containing visible and thermal information. This paper discusses a system which uses a *HeatWave* scanner together with a robot mobile manipulator, allowing these scans to be collected without user operation. An algorithm for this system is also presented, obtaining the positions that the robot mobile manipulator must move to, to find the best view of the given object. A prior model is built up with no knowledge of the scene, and is then improved with a next best view algorithm. The results discussed are simulations only, presented from situations undertaken in visually different scenes.

1 Introduction

In recent years, there as been a rise in the use of small devices for the construction of 3D object maps. A device known as *HeatWave*, extends this capability, with a technology known as 3D thermography. *HeatWave* allows 3D thermal maps of an environment to be created in real-time. With applications in fields such as manufacturing, medical and energy to name a few, 3D models with embedded temperature information are a valuable asset. Current devices, such as *HeatWave*, are handheld, restricting them to be used by a human operator. Attaching such a device to a mobile robot manipulator would allow the user to operate *HeatWave* remotely, or automate the process to increase the efficiency of scans. In this paper we introduce this combined platform, called NeoHeatWave (NeoHW) (Figure 1).

Figure 1: The NeoHeatWave (NeoHW) platform.

This platform allows for movement of both the robotic base and manipulator, giving increased flexibility in the available scanning configurations. Robotic applications in the remote and dangerous fields are potential scenarios of use for this platform. This paper proposes a method of capturing scan with *HeatWave* and a mobile robot manipulator. The proposed algorithms are outlined, and then demonstrated with simulated results. These simulations, can then be integrated and implemented on the

*Corresponding author

robotic platform.

This paper addresses a next-best-view (NBV) planning problem, and proposes an algorithm that generates a set of poses to improve the information in a generated 3D thermal map. This paper is organized as follows. Section 3 presents a brief overview of next-best-view (NBV) path-planning techniques. This is followed by Sections 4 through to 6, explaining the process for initially building a prior model, and then improving upon it with NBV. Section 7 discusses the implication of path planning on the system, and finally Section 8 shows the algorithm implemented in test cases.

2 Background and Related Work

Next-best-view problems (NBV) have commonly been addressed in a broad range of past applications [Connolly and others, 1985]. The term was coined to describe the optimal view required to be selected, when looking at a model or scene. Many types of path planning apply the NBV strategies. Classical and sampling-based are two of these.

Classical path planning involves working primarily in a robot configuration space directly with joint angles [Englot, 2012]. Sampling-based planning involves a similar approach to classical based planning, however rather than obstacles being projected into the configuration space, the robot joints are projected to the workspace instead. From there, the calculations are performed, and then extended back to the configuration space [Elbanhawi and Simic, 2014],[Behnisch *et al.*, 2010]. Another branch of sampling-based planning is known as coverage planning. Many examples of coverage based planning are available from hull detection and reconstruction [Englot *et al.*, 2009],[Walter *et al.*, 2008],[Kumar and Narayanan, 2014] to object reconstruction [Vasquez-Gomez *et al.*, 2014]. These models focus on ensuring that the entire scene is completely covered.

In the NBV algorithms the selection process can be breaked down into several smaller steps [Vasquez-Gomez *et al.*, 2014].

1. Utility Function - ranks views according to their suitability for reconstruction (a mixture of positioning, registration of a new surface and distance)
2. Candidate views - generate views in the robots configuration space
3. Evaluation Strategy - This is decided through a filtration process
4. Stop criterion - Decide when the reconstruction is good enough

For the algorithm proposed in this work, a similar process has been implemented. Moreover, we consider the camera view constrain from *HeatWave*. To consider if an object can be viewed, a model of what the camera can see needs to be created. One such view model, known as a viewsphere, can be implemented to test if an object is viewable from the camera [Ng and Gong, 1999].

3 System Overview

Broken up into several components, the flow chart below in Figure 2 show the NBV algorithm process. Each component, is discussed in detail in individual sections below.

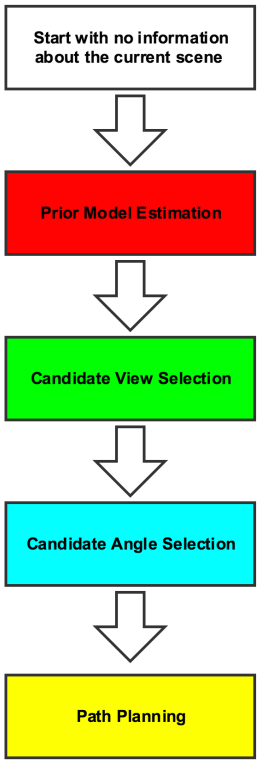

Figure 2: The proposed workflow diagram.

3.1 Selection Constraints

Candidate views generated contain pose information; both orientation and position of the end effector. Position can be represented by a vector containing x, y and z components. The orientation is given by θ and ϕ angles. It is important that these candidate views meet certain criteria, as follows.

1. Workspace Constraint - Views generated must be contained within the robots' workspace. These views can then be executed as they are within reach.
2. Sensor Constraint - The sensor must be able to see the target pose at all times. If it loses sight, it can impact the 3D mapping algorithms functionality. This is also known as a registration constraint.

3. View Constraint - Clusters must be within the field of view of the camera. This will mark them as 'seen', and can be included.
4. Cost Constraint - When multiple points are considered, the cost of transport must be kept to a minimum. Cost of transport for the purposes of this paper, is measured as a distance.
5. Information Constraint - Moving the end-effector position must also generate new information, and add to the existing model.

3.2 Assumptions

In this work we consider the following assumptions.

1. No external obstacles exist in the planned path from the start to end position.
2. The position of the target cluster is known, with respect to the platform.
3. Robot manipulator distance limits have been taken into account.
4. Camera noise and distortion have been ignored.
5. Scene is assumed to be stationary during transit.

3.3 Systems

The NeoHW system is comprised of two main parts, the *HeatWave* 3D thermography device, and the MMO500 platform. As discussed in Section 1, working together these two parts help to tackle some common issues faced when taking 3D scans.

3.4 HeatWave: 3D Thermography System

HeatWave is a lightweight handheld device that delivers accurate real-time temperature information overlaid on a precise 3D model of an object or scene [Moghadam and Vidas, 2014] .

HeatWave consists of a thermal camera, a range sensor and visible camera, all rigidly attached together in an ergonomic form factor (Figure 3).

There are many applications for this, an important one being the medical industry [Moghadam, 2015]. The high frame rate, also allows for real time visualisation and processing [Vidas *et al.*, 2015] [Moghadam *et al.*, 2014]. Figure 4 shows a produced *HeatWave* map for an engine system. Having this temperature information clearly visible to a user is invaluable [Vidas *et al.*, 2013].

3.5 MMO500: Robotic Platform

The MMO500 robotic platform manufactured by Neobotix can be split into two parts - the upper arm, and the lower base. The arm consists of six joints, and when combined, is capable of quite a large variety of movements and motion. The base is moved by the Mecanum wheels, which allow movement both forward and backward, as well as sidewards. Laser scanners are also on both the front and back of the base, for safety and protection.

Figure 3: *HeatWave* 3D thermography device.

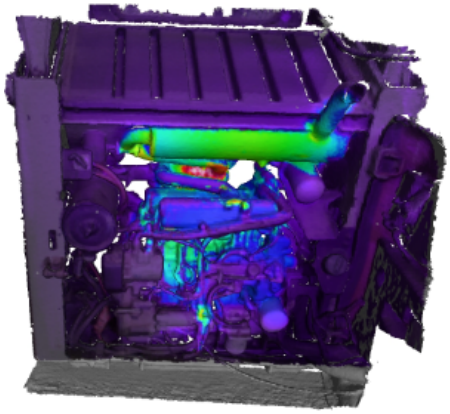

Figure 4: Example of *HeatWave* map.

Figure 5: Robotic MMO500 Platform.

3.6 The Whole System

The *HeatWave* device, connected to the end of the arm, allows the NeoHW to function as one unit (Figure 1).

This paper focuses on algorithms which can be used by this system. Technical parameters and measurements have been based on information for these two devices.

4 Prior Model Construction

Initially, nothing is known about the model. To start examining a scene or object, it is important to have an baseline to take these measurements from. This is known as a prior model. This model is created by generating points in a cone shape around a target position. The points which are within reach of the robot arm (also known as the robot workspace) can then be executed. It is important that the camera is always pointed towards the target point of interest, to ensure that the model is obtained correctly.

The prior model algorithm has initial known parameters which, can be configured,

- t_p - Target Point - the point which the cone is constructed around
- *scan_plane* - Scan Plane - a two dimensional plane on which the scan is performed (ie 'xy')
- *num_strips* - Number of Strips - the number for strips generated for the shape
- *num_waypoints* - Number of Waypoints - the number for waypoints generated in one strip
- $start_offset$ - Starting Offset - the offset which the shape is lifted up from ground level

The output from the algorithm, is a vector V, containing x, y and z. The angle of the sensor is pointed towards the centre, and can be represented by two angles, θ and ϕ. The equations below show how these prior model points were generated. This changes depending on the desired scan plane. For example, the equations below are that for the xy plane. γ varies from 0 to 2π, and strip levels is calculated depending on the number of input strips.

$$xp = t_p.x + cos(\gamma)scale \tag{1}$$

$$yp = t_p.y + sin(\gamma)scale \tag{2}$$

$$zp = t_p.y + strip_levels \tag{3}$$

$$\phi = -tan^{-1}(\frac{yp - t_p.y}{zp - t_p.z}) - \frac{\pi}{2} \tag{4}$$

$$\theta = \frac{\pi}{2} - tan^{-1}(\frac{yp - t_p.y}{xp - t_p.x}) \tag{5}$$

Figures 6 and 7, show a representation of the points generated in 3D space, for two different planes. The red point in the centre is the target point, in which the waypoints have been constructed around.

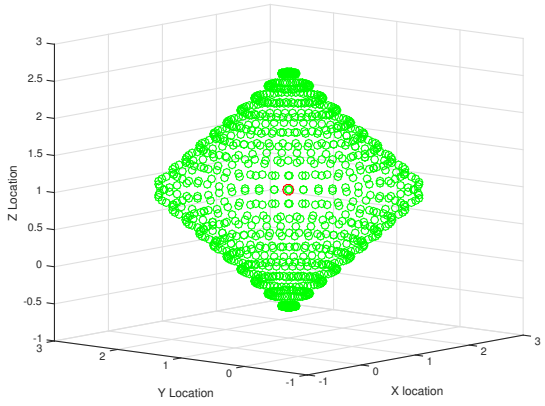

Figure 6: Example of prior model distribution - xy plane.

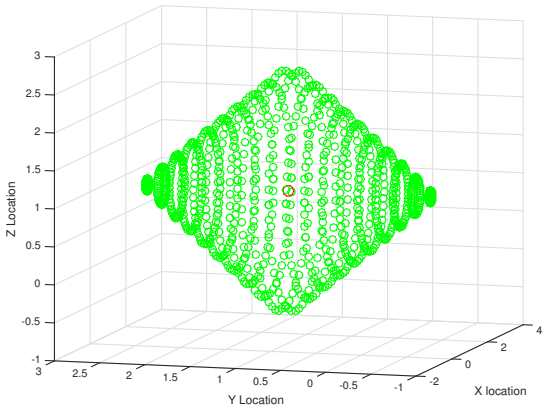

Figure 7: Example of prior model distribution - zx plane.

5 Next Best View - Candidate View Selection

After generating points, and executing them, an initial model of the scene exists. This model will have a varying level of confidence contained in it, depending on the scene. The information extracted from the initial scene is stored in what is known as clusters. These clusters are groups of points, and represent parts of a scene.

Each individual cluster can be represented as a 7-tuple, providing vital information. The tuple is of the form,

$$C = [x_c, y_c, z_c, \theta_c, \phi_c, c_c, w_c] \tag{6}$$

Each element its important for the decision making process. These are defined as follows,

- x_c - The x location of the cluster (in metres)
- y_c - The y location of the cluster (in metres)
- z_c - The z location of the cluster (in metres)

- θ_c - The x - y plane surface normal angle of the cluster (in radians)
- ϕ_c - The x - z plane surface normal angle of the cluster (in radians)
- c_c - The confidence of a cluster (ranging from 0 to 255)
- w_c - The weighting on a cluster (If it is considered more important than another cluster)

Before beginning, the clusters are sorted by their confidence. The clusters with the lowest confidence are placed on top of the list. This means that the order in which points are processed, is the lower confidence points first.

The first critical step is to define the workspace available. Using a robotic arm, the workspace available can be modelled as a circle with a centre point, and radius. This will be referred as R_r (Robot radius (in meters)), and R_c (Robot centre point).

For the purposes of these demonstrates, the workspace will be modelled as a unit sphere. This is due to the physical 1 metre reach of the arm. Represented mathematically by the equation of a circle, the outer edge of the workspace is,

$$(x - R_{cx})^2 + (y - R_{cy})^2 + (z - R_{cz})^2 = R_r^2 \quad (7)$$

After introducing the robot workspace, the clusters are then represented in the scene. Each cluster is modelled as a circle sector, originating from a cluster centre point. The radius of this circle is decided by the characteristics of the sensor used. As well as having an outer radius, the cluster also has an inner radius. This represents the minimum distance the object must be from the sensor. The incident angle is also important to take into account. Ideally, for the most accurate view, the incident angle should be the same as the normal from the surface. There is however a range of acceptable incident angles around the normal angle.

- R_n - Near field of view (in metres)
- R_f - Far field of view (in metres)
- γ_r - Field of view range (in radians)

Using the parameters specified above, the clusters can be modelled as circle sectors, allowing the workspace and clusters to be seen within the same frame. An example frame of what this looks like in a single plane can be seen below,

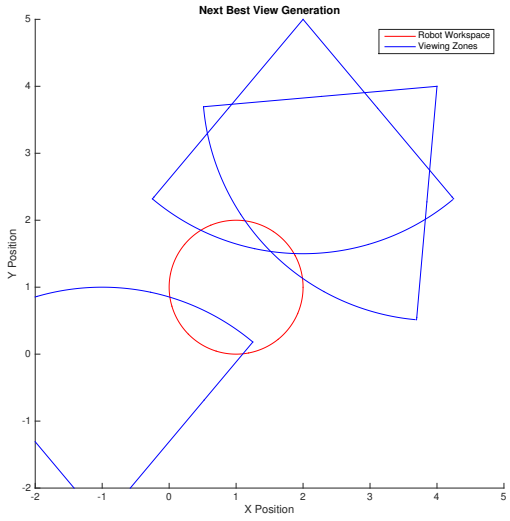

Figure 8: Workspace and Clusters in 2D

With the workspace and cluster viewing areas outlined on the same frame, it can be seen that the areas where the workspace overlaps are the ones that are of interest.

By turning the workspace into many small discrete points, tests can be performed to ensure that points chosen lie both within the workspace, and with the cluster regions. The following conditions will test if a candidate view point is able to see one or more clusters:

X-Y Plane

$$R_f > \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2} \quad (8)$$

$$tan^-1\left(\frac{y_p - y_c}{x_p - x_c}\right) >= \theta_c - \gamma_r \quad (9)$$

$$tan^-1\left(\frac{y_p - y_c}{x_p - x_c}\right) <= \theta_c + \gamma_r \quad (10)$$

$$R_n < \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2} \quad (11)$$

X-Z Plane

$$R_f > \sqrt{(x_p - x_c)^2 + (z_p - z_c)^2} \quad (12)$$

$$tan^-1\left(\frac{z_p - z_c}{x_p - x_c}\right) >= \phi_c - \phi_r \quad (13)$$

$$tan^-1\left(\frac{z_p - z_c}{x_p - x_c}\right) <= \phi_c + \phi_r \quad (14)$$

$$R_n < \sqrt{(x_p - x_c)^2 + (z_p - z_c)^2} \quad (15)$$

Y-Z Plane

$$\frac{(y_p - y_c)}{R_f} + \frac{(z_p - z_c)}{R_f} <= 1 \quad (16)$$

$$\frac{(y_p - y_c)}{R_n} + \frac{(z_p - z_c)}{R_n} >= 1 \tag{17}$$

A list now exists of points which lie within the robot workspace, as well at least one of the cluster sectors. By sorting this list, the points which intersect the most clusters are able to be extracted first, and eliminated, leaving the points which can see less clusters. By repeating the process until all the clusters have been observed, a list of candidate view points is obtained.

6 Next Best View - Candidate Angle Selection

After the selection of the viable candidate view points, viewing angles need to be determined for each of these points. These viewing angles need to ensure that they are able to observe the cluster in question.

Given any number of normal angles returning from clusters to the candidate view point, it is important to choose the incident angle which is best for all clusters. If only one cluster is present for a view point, the most optimal viewing position is when the incident angle is minimised. Representing both the surface normal and viewing angle as vectors, the minimum angle between these can be found by looking at the dot product.

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \ \|\mathbf{B}\| \cos\theta \tag{18}$$

Since the vectors are of equal magnitude, the component of interest is the cosine part. The angle between a surface normal of a cluster and the viewing angle can be represented as: $\theta_{CX} = \theta_x - \theta_c$, where θ_x is the desired observation angle, and θ_c is the surface normal of the cluster. By adding the separate cosine functions together, and finding the maximum of this function (when the angle is minimum), the most optimal angle for viewing the cluster can be calculated. This can be written as,

$$\sum_{n=1}^{i} \cos(\theta_x - \theta_c) \sin(\phi_x - \phi_c) \tag{19}$$

Maximising this function, the most angle with the sum of the least deviation to viewed clusters can determined.

7 Path Planning

After determining the candidate view position, and candidate view angle, the candidate poses for the scene now exist. It is now up to the path planner, to decide how these poses are used.

The first decision for the path planner module to make is to decide how many poses to execute. Depending on the given scene, quite a large number of poses could be returned. It is the path planners task to choose which ones to execute. This could be either choosing points of low confidence, points which see a greater number of clusters, or points which are weighted higher.

The path planner must then decide the order in which to move between these poses. Many path planners have options to optimise the trajectory.

The final consideration for the path planning module is to ensure that during transition between views, the sensor is facing towards the target point. This is the case for *HeatWave* where it needs to face the object of interest, and to not lose track.

The path planning module is an important part of the full NBV process, and is something that is yet to be implemented.

8 Analysis and Discussion

For analysis purposes, four different cases with pieces of data were chosen. The scenarios were selected carefully to test capabilities of the system, with varying amounts of data, demonstrating functionality and possible flaws within the logic of the code.

Each case contains the following parameters which have been set,

- X, Y, Z location (in meteres).
- ϕ angle (in degrees).
- θ angle (in degrees).
- Confidence value (between 1 and 255).

The output figure from each case shows,

- The robot workspace.
- Candidate views marked in red.
- Candidate angles marked in green.
- Clusters marked in black.

Each case is modelled in subsections below.

8.1 Case One

Case one was modelled as two test points, both within viewable range of each other. The algorithm is expected to choose a point and angles which allow both points to be seen. One point was chosen to be generated for this test. Parameters chosen for this case can be seen in Table 1.

Table 1: Case one

Cluster Number	x, y, z location	ϕ Angle	θ Angle	Confidence
1	(4,9,1)	0	-45	1
2	(4.5,9,1.5)	0	-45	1

Case one shows that when presented with two points close together, the NBV algorithm choose a pose which encompasses both cluster centres.

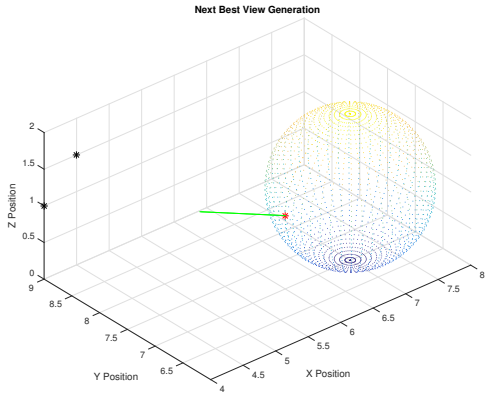

Figure 9: Case one simulation.

8.2 Case Two

Case two was modelled as two test points. These test points were not in viewable range of each other, and point 1 has a higher confidence. The algorithm is expected to choose a pose in which the viewpoint with the least confidence can be seen from. Parameters chosen for this case can be seen in Table 2.

Table 2: Case two

Cluster Number	x, y, z location	ϕ Angle	θ Angle	Confidence
1	(9,5.5,1)	180	135	200
2	(10.5,5,1.5)	180	135	1

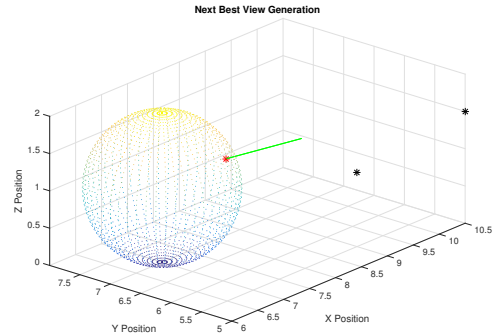

Figure 10: Case two simulation.

Case two shows a similar situation a case one, a view chosen which can see both clusters.

8.3 Case Three

Case three shows an example where 2 clusters can be seen from the same view point, however cannot be seen with the same view angle. Parameters chosen for this case can be seen in Table 3.

Table 3: Case 3

Cluster Number	x, y, z location	ϕ Angle	θ Angle	Confidence
1	(4,10,1)	0	-45	200
2	(4.5,5,2)	0	45	1

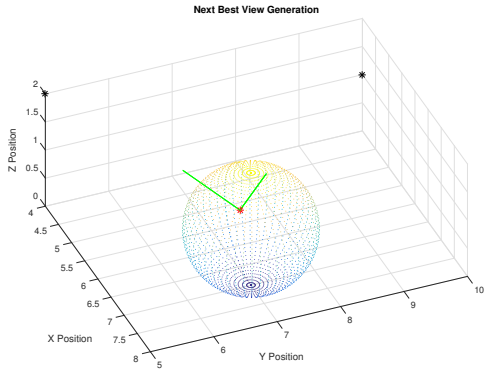

Figure 11: Case three simulation.

Case three essentially presents two options, since all three clusters cannot be seen at once. One view position and two view angles have been selected to observe this scene.

8.4 Case Four

Case four shows another split of clusters. This time with two distinct groups. These points were chosen to demonstrate groups which could not be observed in one view (Table 4).

Figure 12 shows that two view angles have been chosen from one view point. This covers all specified clusters.

8.5 Case Five

Case 5 shows an example of a large number of points, and how view poses can be calculated for these.

Several view points angles have been calculated for this case.

Table 4: Case 4

Cluster Number	x, y, z location	ϕ Angle	θ Angle	Confidence
1	(4.5,9,1)	0	-45	200
2	(4.5,5,2)	0	45	1
3	(4.5,9,1.5)	0	-45	200

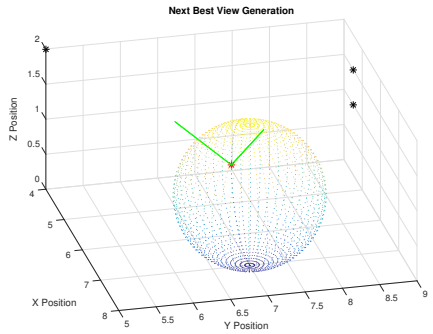

Figure 12: Case four simulation.

Table 5: Case 5

Cluster Number	x, y, z location	ϕ Angle	θ Angle	Confidence
1	(4.5,9,1)	0	-45	1
2	(4.5,5,2)	0	45	1
3	(4.5,9,1.5)	0	-45	1
4	(5,8,1)	0	-30	1
5	(5,8.5,2)	0	-30	1
6	(4,7,2)	0	10	1

9 Conclusions and Future Work

The contribution of this paper is a next-best-view algorithm for coverage based planning. Starting with only a target location, a prior model of the scene can be built up, and then from that improved upon. The NBV algorithm takes into account existing cluster locations as well as a confidence value, to return the optimal points for the manipulator to move to, in order for the sensor to view the scene. The algorithm has been tested in several cases, and performs as expected, choosing an optimal pose. Future work includes implementing the presented algorithm on the robot system, and demonstrating that it works as designed. With this comes some challenges imposed by the physical system. Camera noise and distortion can be taken into this model, to provide a more robust system. The scene in this algorithm has also been assumed to be stationary during movement, this is however not the case. By updated the scene in real time, the candidate points could be changed, increasing efficiency further.

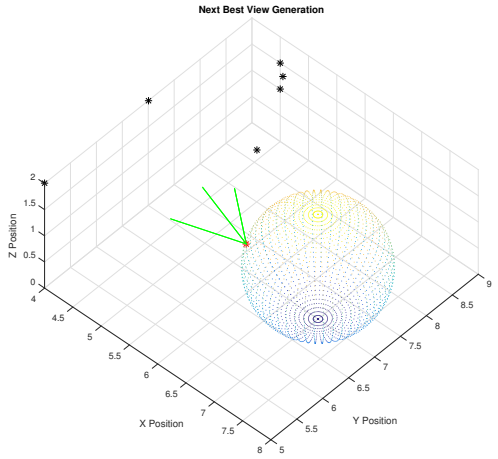

Figure 13: Case five simulation.

References

[Behnisch *et al.*, 2010] Matthias Behnisch, Robert Haschke, and Michael Gienger. Task space motion planning using reactive control. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5934–5940. IEEE, 2010.

[Connolly and others, 1985] Cl Connolly et al. The determination of next best views. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 432–435. IEEE, 1985.

[Elbanhawi and Simic, 2014] M. Elbanhawi and M. Simic. Sampling-based robot motion planning: A review. *Access, IEEE*, 2:56–77, 2014.

[Englot *et al.*, 2009] Brendan Englot, Hordur Johannsson, and Franz Hover. Perception, stability analysis, and motion planning for autonmous ship hull inspection. In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology (UUST)*, 2009.

[Englot, 2012] Brendan J Englot. *Sampling-based coverage path planning for complex 3D structures*. PhD thesis, Massachusetts Institute of Technology, 2012.

[Kumar and Narayanan, 2014] N. Kumar and S.S. Narayanan. Hull detection based on largest empty sector angle with application to analysis of realtime mr images. In *Acoustics, Speech and Signal Processing*

(ICASSP), 2014 IEEE International Conference on, pages 6617–6621, May 2014.

[Moghadam and Vidas, 2014] Peyman Moghadam and Stephen Vidas. Heatwave: the next generation of thermography devices. In *Thermosense: Thermal Infrared Applications XXXVI*, volume 9105. Proc. SPIE, 2014.

[Moghadam *et al.*, 2014] Peyman Moghadam, Stephen Vidas, and Obadiah Lam. Spectra: 3D multispectral fusion and visualization toolkit. In *Proceedings of the Australasian Conference on Robotics and Automation*, 2014.

[Moghadam, 2015] Peyman Moghadam. 3D medical thermography device. In *SPIE Sensing Technology+Applications*, pages 94851J–94851J. International Society for Optics and Photonics, 2015.

[Ng and Gong, 1999] Jeffrey Ng and Shaogang Gong. Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In *ratfg-rts*, page 14. IEEE, 1999.

[Vasquez-Gomez *et al.*, 2014] J.I. Vasquez-Gomez, L.E. Sucar, and R. Murrieta-Cid. View planning for 3D object reconstruction with a mobile manipulator robot. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014.

[Vidas *et al.*, 2013] Stephen Vidas, Peyman Moghadam, and Michael Bosse. 3D thermal mapping of building interiors using an rgb-d and thermal camera. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2311–2318. IEEE, 2013.

[Vidas *et al.*, 2015] Stephen Vidas, Peyman Moghadam, and Sridha Sridharan. Real-time mobile 3D temperature mapping. *Sensors Journal, IEEE*, 15(2):1145–1152, 2015.

[Walter *et al.*, 2008] Matthew Walter, Franz Hover, and John Leonard. Slam for ship hull inspection using exactly sparse extended information filters. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1463–1470. IEEE, 2008.