

Next Best View Planning with an Unstructured Representation



Rowan J. Border
Lincoln College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas 2019

Acknowledgements

I was very fortunate to have received an incredible amount of support from my supervisors, colleagues, friends and family during my DPhil and while writing this thesis. I would not have been able to produce this work and thesis without them.

First and foremost I would like to thank my supervisor, Dr. Jonathan Gammell, for the unparalleled wisdom and unwavering support which he provided throughout my DPhil. He was always generous with his time and the invaluable advice he imparted on my ideas and writing, that is hopefully exemplified in this thesis, has enabled me to become a successful researcher. For that I am immensely grateful.

I thank my co-supervisor, Prof. Paul Newman, for his guidance and incredible work creating a vibrant research community in the Oxford Robotics Institute (ORI).

I would like to thank my examiners, Prof. Nick Hawes and Prof. Patric Jensfelt, for considering this thesis and providing valuable feedback on my work. My thanks also go to Dr. Maurice Fallon for his feedback on my research during the DPhil.

Many thanks to Prof. Niki Trigoni and the members of her research group for their generous provision of lab space and support when conducting the real world experiments presented in this thesis. My thanks to Eileen Westwig and the Oxford University Museum of Natural History (OUMNH) for loaning me the *crocodylus porosus* skull and *rhinoceros sondaicus* pelvis specimens used in these experiments.

It would not have been possible to conduct these experiments and many other field trials without the terrific engineering support provided by Matt Towlson, Peter Get and the rest of the Engineering team in ORI. I thank them for all their hard work creating the sensor systems used in these experiments. My thanks also go to all the members of the Software team in ORI for their great assistance over the years.

The opportunity for me to pursue a DPhil at Oxford was made possible by the generous financial support provided by the Rhodes Scholarships and the CDT in Autonomous Intelligent Machines and Systems (AIMS). It was a privilege to have had this opportunity and become a part of these wonderful communities. My thanks to Mary Eaton and Wendy Poole for all their support during my time at Oxford.

I owe the preservation of my sanity during the DPhil to the phenomenal support of my friends and family. I would like to thank my labmates for their camaraderie and my friends, Abi, Billy, Fran, Gavin, Jamie and Stephen, for being there to remind me of the joys of life. Most importantly, I could not have achieved this without the love and support of my extraordinary parents, Alistair and Karen, and sister, Gemma. Thank you for always being there for me and empowering me to pursue my dreams.

Abstract

High-quality observations of the real world are crucial for creating realistic scene imitations and performing structural analysis. Observations can be used to produce 3D printed replicas of small-scale scenes (e.g., a toy bunny), conduct inspections of large-scale infrastructure (e.g., a building) or integrated into virtual environments that provide immersive experiences for our entertainment and training robotic systems.

Scenes are observed by obtaining point measurements using a sensor from multiple views. These views can be chosen by a human operator or planned using knowledge of existing measurements or an *a priori* scene model. The challenge of selecting the ‘next’ view of a scene to obtain that will provide the ‘best’ improvement in an observation is known as the Next Best View (NBV) planning problem.

This thesis presents work on NBV planning with a novel unstructured scene representation. In contrast to existing literature on the problem, which typically uses structured representations, an unstructured representation does not impose an external structure on scene observations. There is no reduction in the fidelity of information represented or simplifying assumptions made about the scene structure.

This unstructured representation is used to create the Surface Edge Explorer (SEE), a novel NBV planning approach. Observed points are classified based on the local measurement density. Views are chosen to improve the surface coverage of an observation until a minimum point density has been attained. Experiments comparing SEE with structured approaches demonstrate that it is able to obtain an equivalent observation quality using fewer views and a lower computation time.

Novel point-based techniques for considering occlusions and scene visibility are investigated. This work overcomes the raycasting constraints of existing methods used by structured approaches. The best performing strategies for addressing each of these challenges are integrated with SEE to create SEE++. An experimental comparison of SEE++ with SEE and structured approaches demonstrates that it achieves significant improvements in observation performance by requiring fewer views and shorter travel distances while maintaining a reasonable computation time.

Observations of real world scenes using SEE and SEE++ illustrate the successful transference of their capabilities from a simulation environment to the real world. Qualitative results show that both approaches are able to obtain highly complete observations of several scenes with varying size and structural complexity using multiple sensor modalities. Quantitative results demonstrate that SEE++ observes the scenes with greater efficiency than SEE by utilising an increased computational time.

Contents

List of Figures	vii
List of Tables	xiii
Notation	xiv
1 Introduction	1
2 Background	9
2.1 The Next Best View Planning Problem	10
2.1.1 Representing Scene Information	12
2.1.2 Proposing Views	13
2.1.3 Selecting a Next Best View	15
2.1.4 Terminating Observations	16
2.2 Model-based Approaches	16
2.3 Global Representations	19
2.4 Volumetric Representations	21
2.4.1 Sampling Views from a Surface	22
2.4.2 Sampling Views with Path Planning	26
2.4.3 Proposing Views using Scene Information	29
2.5 Surface Representations	31
2.6 Combined Representations	35
2.7 Discussion	39
3 Planning Next Best Views with an Unstructured Representation	44
3.1 Existing Methods	46
3.2 The Surface Edge Explorer (SEE)	47
3.2.1 Point Classification	49
3.2.2 Surface Geometry Estimation	52
3.2.3 View Proposals	54
3.2.4 Next Best View Selection	56
3.2.5 View Adjustment	57
3.2.6 Completion	61

3.3	Evaluation	61
3.3.1	Simulated Sensors	65
3.3.2	View Constraints	65
3.3.3	Algorithm Parameters	66
3.3.4	Performance Metrics	67
3.4	Discussion	67
4	Proactively Handling Occlusions	70
4.1	Existing Methods	71
4.2	Detecting Occlusions	74
4.2.1	Defining Visibility	74
4.2.2	Naive Search	75
4.2.3	Adaptive Search	76
4.3	Proposing Unoccluded Views	77
4.3.1	Representing Occlusions	78
4.3.2	Mean Strategy	79
4.3.3	Eigenvector Strategy	80
4.3.4	Geodesic Strategy	82
4.3.5	Optimisation Strategy	84
4.4	Evaluation	89
4.5	Discussion	89
5	Considering Scene Visibility	96
5.1	Existing Methods	97
5.2	Constructing a Visibility Graph	100
5.2.1	Determining Frontier Visibility	100
5.2.2	Covering Visibility Graph	101
5.2.3	Defining Covisibility	103
5.2.4	Complete Covisibility Graph	104
5.3	Selecting a Next Best View	106
5.3.1	Global Minimum Covisibility	106
5.3.2	Global Maximum Visibility	108
5.3.3	Local Maximum Visibility	109
5.3.4	Local Maximum Visibility-Distance Ratio	110
5.4	Evaluation	112
5.5	Discussion	116

6	Observing Scenes with Fewer Views and Less Travelling	118
6.1	SEE++	120
6.1.1	Updating Occlusions and Visibility	122
6.1.2	Computing a Suitable View Distance	124
6.2	Evaluation	125
6.3	Discussion	125
7	Observing the Real World	132
7.1	Sensor System	135
7.1.1	Intel RealSense D435	135
7.1.2	Velodyne VLP-16	137
7.1.3	Vicon System	138
7.2	Scene Observations	138
7.2.1	Single Box	139
7.2.2	Single Tower	143
7.2.3	Double Towers	146
7.2.4	Small Bookshelf	150
7.2.5	Rhinoceros Pelvis	153
7.2.6	Crocodile Skull	157
7.2.7	Summary	161
7.3	Evaluation	163
7.4	Discussion	169
8	Conclusion	170
8.0.1	Contributions	174
8.0.2	Future Work	175
Appendices		
A	Real World Observations	177
References		204

List of Figures

1.1	Images of consumer and industrial products used to obtain observations of real world objects.	2
1.2	Images of observations obtained for large-scale outdoor scenes. . . .	3
1.3	Images of real world observations that are integrated into simulation environments.	4
2.1	Cross sectional illustrations of different scene representations.	13
2.2	Cross sectional illustrations of different methods for proposing views. .	14
3.1	An illustration of the density-based classification approach used by SEE.	50
3.2	An illustration of the local surface geometry estimate defined by an orthogonal set of vectors.	53
3.3	An illustration of the method for proposing views.	55
3.4	An illustration of the method for adjusting views.	59
3.5	An experimental comparison of SEE with the evaluated volumetric approaches.	62
3.6	An experimental comparison of SEE with the evaluated volumetric approaches (continued).	63
4.1	An illustration of how the visibility of a frontier point from its associated view can be evaluated exactly.	74
4.2	An illustration of how frontier visibility is determined using the naive search-based approximation.	75
4.3	An illustration of a motivating scenario for using an adaptive offset with the occlusion search and how a suitable offset is found.	77
4.4	A cross-sectional illustration of the spherical projection used to represent sight lines along which the observation of a frontier point will be occluded.	78
4.5	An experimental comparison of the sight lines proposed using the mean strategy.	80
4.6	An experimental comparison of the sight lines proposed using the eigenvector strategy.	81

4.7	An experimental comparison of the sight lines proposed using the geodesic strategy.	84
4.8	Experimental results demonstrating a failure case of the optimisation strategy.	88
4.9	Experimental results demonstrating the success of the optimisation strategy.	88
4.10	An experimental comparison of SEE with the presented occlusion strategies.	90
4.11	An experimental comparison of SEE with the presented occlusion strategies (continued).	91
5.1	An illustration of the covering visibility graph representation.	102
5.2	Illustrations of the different covisibility relationships that can exist between views.	104
5.3	An illustration of the complete covisibility graph representation.	105
5.4	An illustration of the Global Minimum Covisibility (GMC) metric for selecting next best views.	107
5.5	An illustration of the Global Maximum Visibility (GMV) metric for selecting next best views.	108
5.6	An illustration of the Local Maximum Visibility (LMV) metric for selecting next best views.	109
5.7	An illustration of the Local Maximum Visibility-Distance Ratio (LMR) metric for selecting next best views.	111
5.8	An experimental comparison of SEE with the presented next best view selection metrics.	113
5.9	An experimental comparison of SEE with the presented next best view selection metrics (continued).	114
6.1	Cross-sectional illustrations of different search methods for selecting view proposals.	123
6.2	A cross-sectional illustration of the method for computing a suitable view distance.	124
6.3	A comparison of SEE++ with SEE and the volumetric approaches.	126
6.4	A comparison of SEE++ with SEE and the volumetric approaches (continued).	127
6.5	A statistical analysis of the observation performance of SEE and SEE++.	129
7.1	The sensor system used to observe the real world scenes.	136
7.2	Photographs of the single box scene.	140

7.3	RealSense pointcloud results for the single box scene.	140
7.4	Velodyne pointcloud results for the single box scene.	141
7.5	Photographs of the single tower scene.	144
7.6	RealSense pointcloud results for the single tower scene.	144
7.7	Velodyne pointcloud results for the single tower scene.	145
7.8	A photograph of the double towers scene.	147
7.9	RealSense pointcloud results for the double towers scene.	147
7.10	Velodyne pointcloud results for the double towers scene.	148
7.11	Photographs of the small bookshelf scene.	151
7.12	RealSense pointcloud results for the small bookshelf scene.	151
7.13	Velodyne pointcloud results for the small bookshelf scene.	152
7.14	Photographs of the rhinoceros pelvis scene.	155
7.15	RealSense pointcloud results for the rhinoceros pelvis scene.	155
7.16	Velodyne pointcloud results for the rhinoceros pelvis scene.	156
7.17	Photographs of the crocodile skull scene.	158
7.18	RealSense pointcloud results for the crocodile skull scene.	158
7.19	Velodyne pointcloud results for the crocodile skull scene.	159
7.20	A quantitative evaluation of SEE and SEE++ for the real world scene observations using the RealSense.	164
7.21	A quantitative evaluation of SEE and SEE++ for the real world scene observations using the RealSense (continued).	165
7.22	A quantitative evaluation of SEE and SEE++ for the real world scene observations using the Velodyne.	166
7.23	A quantitative evaluation of SEE and SEE++ for the real world scene observations using the Velodyne (continued).	167
A.1	Photographs of the single box scene. It contains a 0.6x0.6x0.6m foam box with a unique numerical marking on each side in reflective black duct tape.	178
A.2	The RGB coloured pointcloud result obtained from an observation of the single box scene using SEE with the Intel RealSense D435.	179
A.3	The RGB coloured pointcloud result obtained from an observation of the single box scene using SEE++ with the Intel RealSense D435.	180
A.4	The pointcloud result obtained from an observation of the single box scene using SEE with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each corner of the scene.	181
A.5	The pointcloud result obtained from an observation of the single box scene using SEE++ with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each corner of the scene.	182

A.6	Photographs of each side of the single tower scene. It contains two foam boxes in a stacked configuration with a 45° rotational offset between the boxes.	183
A.7	The RGB coloured pointcloud result obtained from an observation of the single tower scene using SEE with the Intel RealSense D435.	184
A.8	The RGB coloured pointcloud result obtained from an observation of the single tower scene using SEE++ with the Intel RealSense D435.	185
A.9	The pointcloud result obtained from an observation of the single tower scene using SEE with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each corner of the scene.	186
A.10	The pointcloud result obtained from an observation of the single tower scene using SEE++ with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each corner of the scene.	187
A.11	Photographs of each side of the double towers scene. It contains two stacks of rotationally offset foam boxes separated by a distance of less than 1 m.	188
A.12	The RGB coloured pointcloud result obtained from an observation of the double towers scene using SEE with the Intel RealSense D435.	189
A.13	The RGB coloured pointcloud result obtained from an observation of the double towers scene using SEE++ with the Intel RealSense D435.	190
A.14	The pointcloud result obtained from an observation of the double towers scene using SEE with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each side of the scene.	191
A.15	The pointcloud result obtained from an observation of the double towers scene using SEE++ with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each side of the scene.	192
A.16	Photographs of the small bookshelf scene. It consists of two shelves full of books. The photographs show front, back and two side views of the bookshelf.	193
A.17	The RGB coloured pointcloud result obtained from an observation of the small bookshelf scene using SEE with the Intel RealSense D435.	194
A.18	The RGB coloured pointcloud result obtained from an observation of the small bookshelf scene using SEE++ with the Intel RealSense D435.	195

A.19	The pointcloud result obtained from an observation of the small bookshelf scene using SEE with the Velodyne VLP-16. The images use a y -coordinate colourmap and show views of the front, back and two side angles of the bookshelf.	196
A.20	The pointcloud result obtained from an observation of the small bookshelf using SEE++ with the Velodyne VLP-16. The images use a y -coordinate colourmap and show views of the front, back and two side angles of the bookshelf.	197
A.21	Photographs of the rhinoceros pelvis scene. It contains the pelvis of a Javan rhinoceros (<i>rhinoceros sondaicus</i>) specimen that was loaned from the Oxford University Museum of Natural History (OUMNH 19164). The scene exhibits detailed visual features and surface texture with relatively complex geometry.	198
A.22	The RGB coloured pointcloud result obtained from an observation of the rhinoceros pelvis scene using SEE with the Intel RealSense D435.	199
A.23	The RGB coloured pointcloud result obtained from an observation of the rhinoceros pelvis scene using SEE++ with the Intel RealSense D435.	199
A.24	The pointcloud result obtained from an observation of the rhinoceros pelvis scene using SEE with the Velodyne VLP-16. The top row of images use a y -coordinate colourmap and the bottom row of images use a z -coordinate colourmap.	200
A.25	The pointcloud result obtained from an observation of the rhinoceros pelvis using SEE++ with the Velodyne VLP-16. The top row of images use a y -coordinate colourmap and the bottom row of images use a z -coordinate colourmap.	200
A.26	Photographs of the crocodile skull scene. It contains the skull of a saltwater crocodile (<i>crocodylus porosus</i>) specimen that was loaned from the Oxford University Museum of Natural History (OUMNH 19149). The jaw was propped open using a tripod to enable the observation of interior surfaces.	201
A.27	The RGB coloured pointcloud result obtained from an observation of the crocodile skull scene using SEE with the Intel RealSense D435.	202
A.28	The RGB coloured pointcloud result obtained from an observation of the crocodile skull scene using SEE++ with the Intel RealSense D435.	202
A.29	The pointcloud result obtained from an observation of the crocodile skull scene using SEE with the Velodyne VLP-16. The top row of images use a y -coordinate colourmap and the bottom row of images use a z -coordinate colourmap.	203

A.30 The pointcloud result obtained from an observation of the crocodile skull using SEE++ with the Velodyne VLP-16. The top row of images use a y -coordinate colourmap and the bottom row of images use a z -coordinate colourmap.	203
--	-----

List of Tables

3.1	Observation performance metrics for SEE and the evaluated volumetric approaches.	64
3.2	The field-of-view in degrees, θ_x and θ_y , and resolution in pixels, w_x and w_y , of the simulated depth sensors used to obtain observations of the scene models.	65
3.3	The parameters used by SEE and the volumetric approaches to observe the one-metre standard models and the 40 metre model of the Radcliffe Camera.	66
4.1	Observation performance metrics for SEE and the presented occlusion strategies.	92
4.2	A statistical analysis of the observation performance of SEE and the presented occlusion strategies.	94
5.1	Observation performance metrics for SEE and the presented next best view selection metrics.	115
5.2	A statistical analysis of the observation performance of SEE and the presented next best view selection metrics.	116
6.1	Observation performance metrics for SEE, SEE++ and the volumetric approaches.	128
6.2	A statistical analysis of the observation performance of SEE and SEE++.	130
7.1	Observation performance metrics for the single box scene.	141
7.2	Observation performance metrics for the single tower scene.	145
7.3	Observation performance metrics for the double towers scene. . . .	148
7.4	Observation performance metrics for the small bookshelf scene. . . .	152
7.5	Observation performance metrics for the rhinoceros pelvis scene. . .	156
7.6	Observation performance metrics for the crocodile skull scene. . . .	159

Notation

General Notation

- a Scalars are denoted by lower-case unbolded variables.
- A Sets are denoted by upper-case unbolded variables.
- $\mathbf{a}$ Vectors are denoted by lower-case bolded variables.
- $\mathbf{A}$ Matrices are denoted by upper-case bolded variables.

Specific Symbols

- $\emptyset$ Denotes the empty set.
- $\mathbf{I}$ Denotes the identity matrix.
- $\mathbf{f}$ Denotes a frontier point.
- $\mathbf{v}$ Denotes a view proposal.
- $\mathbf{x}$ Denotes a view position.
- ϕ Denotes a view orientation.
- r Denotes the resolution radius.
- ρ Denotes the target measurement density.
- d Denotes the view distance.
- ψ Denotes the occlusion search distance.
- τ Denotes the view visibility update limit.

Specific Operators

- $|\cdot|$ Denotes the cardinality of a set.
- $\|\cdot\|$ Denotes the L²-norm of a vector.

1

Introduction

Humans have always strived to capture *observations* of the real world that are indistinguishable from our own perceptions. It is only relatively recently that this capability has entered the realm of possibility. The invention of photography in the 19th century made it possible to capture a realistic observation in two dimensions. The first three-dimensional observations were obtained using manual photogrammetry in the early 20th century to create terrain elevation maps from aerial photographs. Our ability to capture high-quality 3D observations of the real world has advanced rapidly since the advent of digital technology, to the extent that it is now possible to observe small-scale scenes using consumer products (e.g., the MakerBot Digitizer; Fig. 1.1a).

Realistic observations are crucial for the accurate analysis and imitation of the real world with digital systems. High-accuracy scanners attached to industrial robots are used to compare the structure of manufactured parts with ground truth production models for quality control (e.g., the MetraSCAN 3D-R; Fig. 1.1b). Observations obtained from surveying large-scale outdoor structures, typically with an aerial platform, can be used for infrastructure inspection (Fig. 1.2a) or to preserve edifices of historical significance. For example, observations of Notre Dame Cathedral (Fig. 1.2b) and the ancient city of Palmyra, destroyed by ISIS, are being used to aid their respective reconstruction efforts.

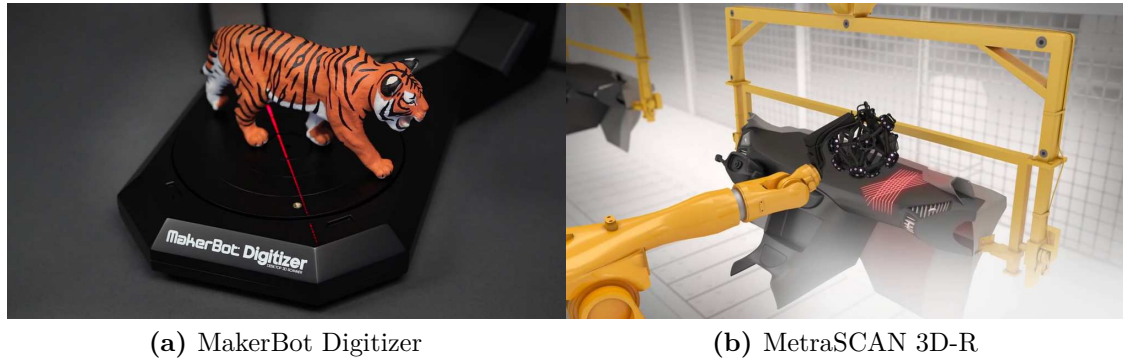
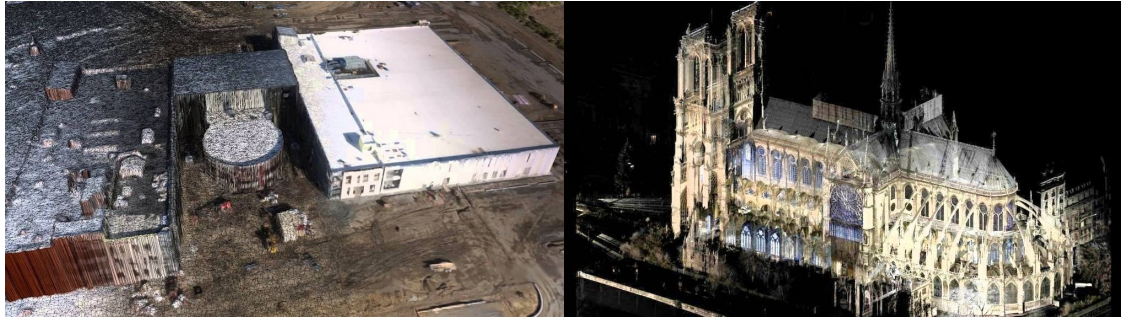


Figure 1.1: Images of consumer and industrial products used to obtain observations of real world objects. (a) shows the MakerBot Digitizer, a desktop 3D scanner that captures observations of small objects placed on a rotating turntable using laser measurements. Image courtesy of MakerBot. (b) shows a MetraSCAN 3D-R, a LiDAR sensor array on an industrial robot arm which obtains accurate observations of manufactured parts using known production models. Image courtesy of Creaform.

High-quality observations are also being used to improve the realism of virtual environments for the purposes of entertainment and testing robotic systems. Observations of real structures are integrated into video games to provide an immersive experience (e.g., the Sphinx in *Assassin's Creed: Origins*; Fig. 1.3a). Accurate visual and structural observations of the real world can improve the fidelity of virtual testing environments for robotic systems and reduce the reality gap for physical deployment (e.g., *Carcraft* from Waymo; Fig. 1.3b). As the capability of 3D sensors improves it is becoming possible to create increasingly realistic simulations of the real world for such applications. Whether it is possible to achieve realism that is indistinguishable from our reality (i.e., the simulation hypothesis) is a hotly debated topic, and beyond the scope of this thesis.

Capturing high-quality observations is a challenge regardless of their final purpose. A *scene* (i.e., a bounded region of space) is observed by capturing individual point *measurements* from surfaces using a 3D *sensor*. These points are captured by estimating the distance of scene surfaces along a set of rays originating from the sensor position. This distance can be computed by triangulating the relative positions of unique visual features in multiple images (i.e., a stereo camera), evaluating the deformation of a known pattern projected over scene surfaces (i.e., an

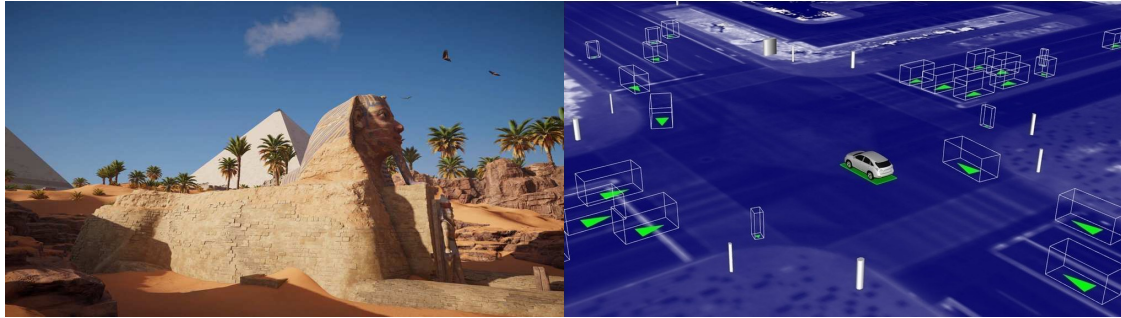


(a) Aerial Observation using DroneDeploy (b) Observation of Notre Dame Cathedral

Figure 1.2: Images of observations obtained for large-scale outdoor scenes. (a) shows the observation of an industrial site obtained using DroneDeploy, a system for surveying structures using photographs captured with an aerial platform. Image courtesy of DroneDeploy. (b) shows an observation of Notre Dame Cathedral obtained by capturing measurements from multiple views with a Leica ScanStation C10 and coloured using panoramic photographs. Image courtesy of Andrew Tallon.

RGB-D camera) or measuring the time-of-flight of an emitted infrared light pulse by detecting its reflection (i.e., a LiDAR sensor). The result is a set of 3D points with x - and y -coordinate positions defined by the distribution of rays within the sensor field-of-view and z -coordinate positions given by the computed depth values.

Observations are obtained by combining point measurements captured from multiple *views* (i.e., sensor poses) around a scene. An observation can be considered complete if the measurements obtained provide coverage of all visible scene surfaces. The coverage achieved depends on the capabilities of the sensor used, the scene structure and the views from which measurements are obtained. The views can be chosen by a human operator, either by specifying a predefined sequence of views or by iterative selection based on an empirical consideration of the current observation state. Using a predefined sequence of views is typically not a successful strategy for obtaining complete observations of different scene structures as the coverage obtained from a view varies with the surface geometry being observed. Relying on the empirical selection of views by a human operator is often not desirable and may not be possible in some circumstances. The quality of observations obtained depends on the operator’s expertise and the allowance for human-in-the-loop control of the sensor platform, which may not be possible when operating in certain environments.



(a) The Sphinx in Assassin's Creed: Origins

(b) Carcraft Simulation Environment

Figure 1.3: Images of real world observations that are integrated into simulation environments. (a) shows the Sphinx in Assassin's Creed: Origins, which is based on a real observation using photogrammetry. Image courtesy of Ubisoft. (b) shows the real world observations included in the Carcraft simulation environment developed by Waymo for training autonomous driving systems. Image courtesy of Waymo.

A better solution for selecting views, which mitigates human uncertainty, is to formulate an algorithm that chooses views by evaluating *a priori* scene knowledge and the current state of an observation. The challenge of planning a ‘next’ view based on these considerations that can provide the ‘best’ improvement in a scene observation is known as the Next Best View (NBV) planning problem. This was initially defined in seminal work by Connolly (1985) together with the first solutions.

Approaches to the NBV planning problem can be categorised based on their use of *a priori* scene knowledge. Those that require prior information of the scene structure to plan next best views are referred to as *model-based*. These approaches are capable of obtaining high-quality observations of scenes for which ground truth knowledge is available (e.g., to compare a manufactured part with its production model) but not do generalise to observing unknown scenes. *Model-free* approaches do not require *a priori* scene knowledge to obtain an observation, allowing them to observe unknown scenes. They plan next best views by evaluating information on the current observation state, which is encoded in a given scene *representation*.

Most model-free NBV planning approaches use *structured* representations. Scene information is encoded in an external structure imposed on the point measurements captured from views. *Volumetric* representations segment the scene volume into a 3D voxel grid that encodes information on measurement occupancy and the observation

state. *Surface* representations define a connectivity between point measurements to create a triangulated surface mesh. These representations are often used to simplify the NBV planning problem by considering point measurements in aggregate but this can reduce the quality of observations. The fidelity of scene information considered is limited by the structural resolution, as given by the voxel size for a volumetric representation or the parameters defining mesh connectivity for surface representations. This means that the observation of scene features smaller than the structural resolution can not be ensured when planning next best views.

It is often necessary to use simplifying assumptions to reduce the computational cost of evaluating and updating an external structure. Approaches with a volumetric representation evaluate potential next best views by raycasting their voxel grid. The evaluation cost is typically reduced by sampling a fixed set of views around the scene rather than adaptively proposing views based on captured measurements. This can restrict the achievable scene coverage as it is dependent on the quantity, distribution and orientations of the sampled views. Many approaches also obtain a fixed number of views in lieu of evaluating observation completeness from captured measurements.

Surface representations are often too computationally expensive to update in real-time and therefore such approaches typically use multiple observation stages. An initial mesh is computed from sparse measurements which is then improved by planning a subsequent observation. The views planned are limited by the initial mesh construction and are often not updated during the observation to account for new measurements, reducing the quality of observation achieved. The overall time required to observe a scene is also increased by the multiple observation stages.

This thesis presents work on a novel *unstructured* representation based on the measurement *density* in a scene observation. It is founded on the principle that obtaining a given minimum measurement density on all scene surfaces is a sufficient condition to achieve a complete observation. No external structure is imposed on point measurements so all scene information is represented with a point-based encoding and only local pointwise computations are required to update and evaluate the representation. Points are individually classified based on the local measurement

density rather than being aggregated into an external structure so the fidelity of scene information considered is not limited by a given structural resolution. The measurement density is computed within a specified radius but this only limits the extent of scene information considered and not the fidelity of point measurements.

This unstructured representation is computationally efficient to maintain as only local updates are performed when new measurements are obtained so simplifying assumptions about the scene structure are not required. Views are proposed to improve a scene observation based on information encoded in the representation and can be adapted to account for new measurements. This means that greater improvements in scene coverage can be achieved by directly considering the scene structure. Only a sufficient number of views are captured to achieve the specified measurement density and only a single observation stage is required. This improves efficiency by reducing the number of views, travel distance and computation time.

The remainder of this thesis is organized as follows. Chapter 2 presents a review of the NBV planning problem and relevant literature on existing approaches. The key considerations of a NBV planning approach are described: the scene representation, the method for proposing views, the metric for selecting a next best view and criteria for terminating an observation. The literature review discusses existing approaches in the context of these key considerations.

Chapter 3 presents the Surface Edge Explorer (SEE), a NBV planning approach implementing the novel unstructured density representation. The observation performance of SEE is compared experimentally with approaches using a volumetric representation for the observation of scenes in a simulation environment. This work was first presented in Border et al. (2017) at the 2017 Joint Industry and Robotics CDTs Symposium and extended in Border et al. (2018) at the 2018 IEEE International Conference on Robotics and Automation.

Chapter 4 presents an investigation of point-based strategies for *proactively* handling occlusions with an unstructured representation. Methods for detecting occluding points are discussed and a novel representation for pointwise occlusions

is presented. Several strategies for proposing unoccluded views are investigated and an experimental evaluation with SEE of their observation performance is presented.

Chapter 5 presents an investigation of methods for considering scene visibility with an unstructured representation. Different graphical representations are investigated for encoding information on the visibility of target points from the set of view proposals. Several metrics for selecting next best views using a novel *covisibility* graph are presented. Their effect on observation efficiency is compared using SEE.

Chapter 6 presents SEE++, a NBV planning approach which integrates the best performing techniques for proactively handling occlusions and considering scene visibility using an unstructured representation with SEE to improve the efficiency of scene observations. The performance of SEE++ is compared experimentally with SEE and NBV planning approaches using a volumetric representation for observations of scene models in a simulation environment. This work was presented in Border and Gammell (2020) at the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Chapter 7 presents a real world demonstration of SEE and SEE++ for the observation of scenes with varying structures independently using both a stereo camera and LiDAR sensor. Qualitative pointcloud results obtained from the scene observations are accompanied by a discussion of their completeness and quality. Quantitative metrics are used to evaluate the observation performance of SEE and SEE++. This work is being prepared for submission to a field robotics journal.

The work presented in this thesis makes the following key contributions:

1. A novel unstructured scene representation using measurement density, which is founded on the principle that obtaining a minimum measurement density on all scene surfaces is a sufficient condition to achieve a complete observation.
2. SEE, a NBV planning approach using this novel representation that imposes no assumptions on the scene structure. Views are proposed, selected and adapted to improve a scene observation by directly considering point measurements.

3. An investigation of point-based strategies for proactively handling occlusions which detect occluded views of target points and aim to propose alternative unoccluded views from which the target surfaces can be successfully observed.
4. An investigation of methods for considering scene visibility with an unstructured representation which aim to select next best views that can provide the greatest improvements in scene coverage while travelling short distances.
5. SEE++, a NBV planning approach that integrates the most successful methods for handling occlusions and considering scene visibility with SEE to greatly improve observation performance by utilising an increased computation time.
6. Real world demonstrations of the presented approaches obtaining observations of several different scenes using both a stereo camera and LiDAR sensor.
7. Implementations of SEE and SEE++ that will be made available open-source to aid further research into NBV planning with unstructured representations.

2

Background

Contents

2.1	The Next Best View Planning Problem	10
2.1.1	Representing Scene Information	12
2.1.2	Proposing Views	13
2.1.3	Selecting a Next Best View	15
2.1.4	Terminating Observations	16
2.2	Model-based Approaches	16
2.3	Global Representations	19
2.4	Volumetric Representations	21
2.4.1	Sampling Views from a Surface	22
2.4.2	Sampling Views with Path Planning	26
2.4.3	Proposing Views using Scene Information	29
2.5	Surface Representations	31
2.6	Combined Representations	35
2.7	Discussion	39

This chapter presents an overview of the NBV planning problem (Sec. 2.1) and a review of relevant NBV planning literature. The literature review focuses on NBV planning approaches for scene observation as these are most relevant to the work presented in this thesis. However, the methodology of NBV planning is also applied to tasks other than scene observation in the wider field of active vision (Chen et al. 2011), including for object search (Kunze et al. 2017), recognition (Foissotte et al.

2009; Wu et al. 2015; McGreavy et al. 2017) and reducing measurement uncertainty in multiview stereo reconstructions (Wenhardt et al. 2007; Trummer et al. 2010).

Literature surveys of work on NBV planning for scene observation have assessed the performance of several algorithms and presented different strategies for categorising approaches (Tarabanis et al. 1995; Scott et al. 2003; Karaszewski et al. 2016a). Tarabanis et al. (1995) classify approaches by their method for proposing views. Scott et al. (2003) use a two-dimensional categorisation based on the use of an *a priori* model and the chosen scene representation. Karaszewski et al. (2016a) focus their review on evaluating the performance of various state-of-the-art approaches on multiple scenes using different sensors.

This review adopts the classification scheme presented by Scott et al. (2003) to discuss approaches based on their scene representation and use of an *a priori* model. Model-based approaches are discussed collectively in Section 2.2. The remaining sections discuss model-free approaches grouped by their scene representation. Approaches using a volumetric representation are further subdivided by their method for proposing views. The assessment of each approach considers the use of scene information, the method for proposing views, the metric for selecting next best views and the termination criteria if any are specified.

2.1 The Next Best View Planning Problem

The challenge of NBV planning is proposing and selecting a sequence of views that can obtain the ‘best’ observation of a scene as efficiently as possible. The quality of a scene observation can be quantified by its accuracy (i.e, how closely an observation resembles the actual scene) and completeness (i.e., what proportion of the scene is captured by an observation). The accuracy of an observation primarily depends on the capabilities of the sensor used to obtain measurements but can be improved by considering the scene structure. For example, measurements captured using stereo and RGB-D sensors are typically more accurate when the view orientation is orthogonal to the surface plane. The completeness of an observation depends on the

scene coverage obtained from captured views. This can be improved by considering the visibility of scene volumes or surfaces when proposing and selecting views.

The efficiency of scene observations can be quantified by the sensor travel distance, computational cost and number of views required. It is often necessary to observe scenes with the greatest possible efficiency in order to respect the operational constraints of a sensor platform. Observing scenes with short travel distances reduces the energy consumption of the sensor platform when moving between views. A lower computational cost is desirable when using platforms with limited processing power. Obtaining observations using the fewest number of views necessary to achieve a given quality limits the storage required for sensor measurements and typically also reduces the overall travel distance and computational cost.

This challenge is known as the NBV planning problem as each view is chosen subsequent to evaluating the information obtained from previous views and in some cases *a priori* scene knowledge. A solution to the problem can be formally expressed as a function, $\mathbf{v}_{i+1} = \text{NBV}(W, K)$, which selects a view, $\mathbf{v}_{i+1}$, from a set of potential views, W , based on information obtained from previous views or *a priori* scene knowledge, K . The next best view is selected to provide the greatest improvement in the scene observation by evaluating a function defined from a given set of quantitative metrics, $M(\mathbf{v}, K)$,

$$\mathbf{v}_{i+1} = \text{NBV}(W, K) = \arg \max_{\mathbf{v} \in W} M(\mathbf{v}, K). \quad (2.1)$$

Most NBV planning approaches select views and obtain new measurements until a given termination criterion is satisfied. This can be specified in terms of the observation cost (e.g., a number of views, travel distance or time constraint) or the observation quality (e.g., the completeness or accuracy of the resulting model).

There are many different approaches to the NBV planning problem, all of which present solutions to the common challenges of representing scene information (Sec. 2.1.1), proposing views to observe the scene (Sec. 2.1.2), selecting a next best view (Sec. 2.1.3) and deciding when to terminate an observation (Sec. 2.1.4).

2.1.1 Representing Scene Information

Scene information is extracted from sensor measurements or *a priori* knowledge to inform the proposal of views, the selection of next best views and the termination criteria for completing a scene observation. Sensor measurements are typically a collection of observed points (i.e., a pointcloud) obtained using an RGB-D sensor, stereo camera (e.g., an Intel RealSense D435) or LiDAR (e.g., a Velodyne VLP-16).

Approaches that require the existence of an *a priori* scene model for planning views are referred to as *model-based*. Model-based approaches are useful for comparing an as-built object with a known ground truth (e.g., a manufactured part with its CAD model) but do not generalise to unknown scenes.

Most approaches plan views using incomplete and frequently noisy scene information obtained from sensor measurements. These approaches require no *a priori* scene model and are referred to as *model-free*. Many model-free approaches primarily consider geometric information extracted from a *structured* representation imposed upon the scene observation. Global representations consider observed points to be measurements sampled from an underlying globally connected structure.

Volumetric representations segment the scene volume into a three-dimensional grid known as a *voxel grid* (Fig. 2.1a). The state of each cell in the voxel grid can encode information regarding its visibility (i.e., which views can it be seen from), observation status (i.e., has it been seen before) and occupancy (i.e., does it contain any point measurements). Surface representations connect observed points to create a *surface mesh* (Fig. 2.1b). This mesh can contain information about the distribution of observed points and the boundaries of the current scene observation. Some NBV planning approaches use a combination of volumetric and surface representations for encoding scene information.

Unstructured representations do not impose an external structure on the scene or assume any connectivity between observed points. This thesis presents work on NBV planning using an unstructured *density* representation (Fig. 2.1c). Point measurements are classified based on the local density of neighbouring points within a specified radius. Those with a given minimum density are completely observed

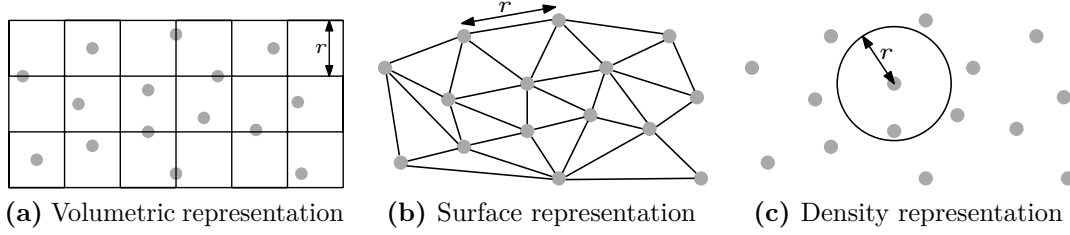


Figure 2.1: Cross sectional illustrations of (a) a volumetric scene representation (i.e., a voxel grid), (b) a surface representation (i.e., a triangulated mesh) and (c) the unstructured density representation presented in this thesis. The resolution parameter, r , defines the voxel size for a volumetric representation, the edge length for a surface representation and the search radius for the density representation.

and those with insufficient density are partially observed. Points along the boundary of these sets are used to propose views that will improve the scene observation.

2.1.2 Proposing Views

Most NBV planning approaches select next best views from a set of views that are proposed to improve the observation of a scene. An ideal set of view proposals would provide complete coverage of a scene while accounting for occlusions, maximising observation quality and minimising computational cost. Views can be obtained from a view surface encompassing the scene, sampled using path planning methods or proposed using information from existing measurements or an *a priori* model (Fig. 2.2).

Approaches for observing scenes with *a priori* models often propose a set of views offline before the scene observation begins. These views can utilise all of the scene knowledge provided by the *a priori* model to obtain the best possible surface visibility by accounting for occlusions and requiring views to be oriented orthogonally to surfaces in order to capture the highest quality measurements.

Obtaining views from a view surface encompassing the scene is computationally efficient as no evaluation of scene information is required (Fig. 2.2a). However, the computational cost of selecting a next best view is often significant as an exhaustive evaluation of every view proposal is performed. High coverage of the scene volume can be achieved by uniformly sampling views on a spherical or hemispherical surface, as constrained by the reachability of the sensor and the scene structure. Views are

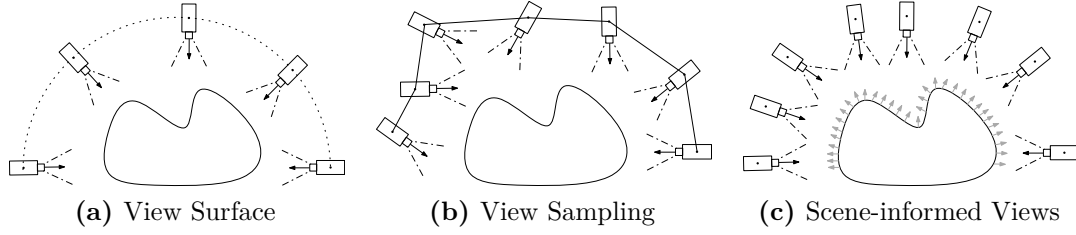


Figure 2.2: Cross sectional illustrations of different methods for proposing views. They can be (a) obtained from a view surface encompassing the scene, (b) sampled in free space around the scene using path planning methods, or (c) proposed based on scene information (e.g., surface normals; grey arrows) obtained from sensor measurements or an *a priori* scene model.

typically oriented towards the centre of the scene volume. The observation quality and surface coverage obtainable is often limited as geometric scene information is not considered. Occlusions can prevent the observation of scene surfaces as views are not adapted to account for restrictions in surface visibility. Measurements with low accuracy may be obtained from views with an acute angle between their orientation and the observed surface geometry when using a stereo camera or RGB-D sensor.

Sampling views of a scene using path planning methods (e.g., RRT; LaValle 1998 or RRT*; Karaman and Frazzoli 2011) is computationally more expensive than sampling them from a surface encompassing the scene but can significantly reduce the computational cost of evaluating view proposals (Fig. 2.2b). View sampling is typically performed locally which reduces the number of view proposals considered when selecting a next best view. Most volumetric approaches using this method do not directly consider scene information when sampling views. Occlusions, surface visibility and observation quality are only considered when evaluating view proposals.

Model-free approaches to NBV planning that do not sample view proposals from a surface or with planning methods propose views based on scene information extracted from sensor measurements (Fig. 2.2c). This results in a dynamic set of view proposals that can change based on information obtained from new measurements. New views can be proposed to observe recently discovered scene surfaces and existing views adjusted to provide better coverage of partially observed surfaces. Occlusions can be handled reactively by incrementally adjusting views until the target surface

becomes visible or proactively detected and avoided. More accurate measurements can be obtained by considering estimates of the surface geometry being observed.

2.1.3 Selecting a Next Best View

A next best view is selected from the set of view proposals. The chosen view is the one that will ‘best’ improve the scene observation according to a set of evaluation metrics, as in (2.1). The most commonly used metrics consider sensor travel distance, scene coverage and observation quality. Most NBV planning approaches combine these metrics to try and obtain the most complete scene observations with the highest accuracy using the fewest views and shortest sensor travel distance.

Reducing the travel distance required to observe a scene is important when considering the energy constraints of mobile robots. Platforms powered by onboard batteries (e.g., UGVs and UAVs) often have limited energy capacity that can be rapidly drained by moving long distances. It is therefore desirable to select views that can obtain a complete scene observation while requiring the least movement.

Scenes can be observed using fewer views by obtaining greater increases in surface coverage per view. This is achieved by evaluating the visibility of incompletely observed surfaces from each view and selecting the view with the best visibility. Requiring fewer views improves the efficiency of most NBV planning approaches as a computational and storage cost is incurred for processing each sensor measurement.

The accuracy of point measurements depends on the sensor capabilities, the scene structure and the view distance of the sensor from scene surfaces. Measurement accuracy can be quantified by considering the variance of observed points and their consistency with previous measurements. When using stereo cameras and RGB-D sensors, higher accuracy measurements can be obtained by observing surfaces from orthogonally oriented views at short distances, as the uncertainty of depth measurements increases with distance and the acuteness of the observation angle.

2.1.4 Terminating Observations

The observation of a scene can be terminated manually or when a specified termination criterion has been satisfied. A termination criterion can be defined in terms of observation cost (e.g., number of views, travel distance or computation time) and observation quality (e.g., surface coverage or measurement accuracy).

NBV planning approaches that rely on manual termination require human intervention and are not deployable on autonomous systems. This limits their use to scenarios where a human operator has the ability to oversee the scene observation in real-time and terminate the NBV planning at a time of their choosing.

Approaches using observation cost as a termination criterion are best suited for deployment on platforms with strict operating constraints. They can guarantee that a scene observation will be obtained within a specified time limit, travel distance or number of views, regardless of the completeness or quality of the resulting model.

Specifying a termination criterion in terms of observation quality ensures that a NBV planning approach will propose, select and obtain new views until it has obtained a sufficient coverage of sensor measurements. It is not possible for an approach to guarantee complete surface coverage or measurement accuracy as these are dependent on the scene structure and sensor capabilities. For example, stereo cameras are often unable to obtain accurate measurements of untextured surfaces.

2.2 Model-based Approaches

Model-based approaches for NBV planning (Tarbox and Gottschlich 1995; Blaer and Allen 2007; Scott 2009; Bircher et al. 2015; Kaba et al. 2017) are typically applied to inspection tasks where a scene observation is obtained to compare the as-built structure of an object with a known ground truth. These approaches require *a priori* scene knowledge for proposing views and selecting a next best view.

Tarbox and Gottschlich (1995) present three approaches for obtaining scene observations with different next best view selection metrics and a known reference model. The objective is to observe a set of points sampled from the reference model

surface. A point is considered observable from a given view if it is unoccluded and the view orientation is sufficiently orthogonal to the local surface of the reference model. Occlusions are detected by raycasting a voxel grid representation of the scene to identify occupied voxels. View proposals are obtained from a sphere encompassing the scene. Sampled points are associated with the set of view proposals from which they are observable. The first approach chooses the view that can observe the most points with small observability sets (i.e., those most difficult to observe). The second approach chooses the view that can observe the greatest number of points from an orientation orthogonal to their respective surfaces. The third approach performs a global optimisation to find the minimum set of views that can observe the set of sampled points. The approaches terminate when all of the sampled points are observed.

Blaer and Allen (2007) present a multistage approach for observing large-scale scenes. The initial stage utilises an *a priori* two-dimensional map of the environment to plan a set of views from which all boundary edges can be observed. Views are randomly sampled in the map and the boundary edges observable within the frustum of each view, as defined by the sensor properties, are identified. The visibility of each edge is further constrained based on the orthogonality of the view orientation. A minimum covering set of views is selected from which all boundary edges can be observed. The secondary observation stage uses a volumetric representation (i.e., a voxel grid) of the model obtained from the initial observation to identify incompletely observed regions. Voxels are associated with a state denoting their observation (i.e., have they been viewed) and occupancy (i.e., do they contain point measurements) status. They can be *observed* or *unobserved* and have an *occupied* or *unoccupied* occupancy state. The set of view proposals contains the centres of unoccupied voxels in the initial model that intersect the ground plane of the *a priori* map. The next best view selection metric chooses the view from which the most unobserved boundary voxels are visible. An unobserved boundary voxel is defined as an unobserved voxel with at least one neighbouring voxel that is both observed and unoccupied. A voxel is visible from a view if no occupied voxels are intersected

by raycasting from the view. A scene observation terminates when the number of visible unobserved boundary voxels falls below a given threshold.

Scott (2009) presents a model-based view planning approach that is applicable to both object inspection and scene observation. Views are proposed to observe surface points sampled from a scene model as presented by Tarbox and Gottschlich (1995). The model does not have to be a known ground truth and can instead be a rough estimate from a initial scene observation obtained using a predetermined view trajectory. This multistage approach is similar to that presented by Blaer and Allen (2007). A surface representation (i.e., a triangulated mesh) is used to define the connectivity between sampled points. A view is proposed to observe each sampled point at a given distance from the point along the normal to the surface mesh. Occlusions are not considered. The observability of every point from each view is evaluated. A point is considered observable from a given view if the point lies within the view frustum, is visible given the sensor properties and if a sufficiently dense and precise set of point measurements can be obtained around the point. A minimum covering set of views is selected to observe the scene. A view trajectory plan is computed using an approximation of the *travelling salesman* problem.

Bircher et al. (2015) present a view planning approach for structural inspection tasks. The objective is to obtain a complete scene observation with the shortest possible travel distance by iteratively planning a view trajectory based on an *a priori* triangulated surface mesh. A view is proposed to observe every triangle in the surface mesh. The pose of each view is optimised to reduce the distance between neighbouring views while maintaining a minimum observation angle with the surface plane and a bounded view distance. Occlusions are not considered. The shortest path between the view proposals is computed as an approximate solution of the travelling salesman problem. The optimised view proposals are updated and a new trajectory is computed after each view which replaces the previous solution if it provides a shorter path. The scene observation terminates when there are no view proposals remaining (i.e., the termination criterion is a view limit defined by the number of triangles in the surface mesh).

Kaba et al. (2017) present a reinforcement learning approach to model-based view planning. This approach requires an *a priori* surface mesh of the scene. A set of views are proposed given this mesh but the process for proposing these views, including whether occlusions are considered, is not discussed. Next best views are selected to maximise the observable area of the mesh relative to an exponential weighting of its perimeter. The weight for selecting each view is learned with a reinforcement learning approach. A scene observation terminates when every triangulated surface in the *a priori* mesh has been observed.

Model-based approaches provide a suitable solution to the problem of obtaining an observation of a known scene for comparison with a ground truth model. However, they do not address the NBV planning problem for observing unknown scenes.

2.3 Global Representations

Model-free approaches to NBV planning with a global representation (Yuan 1995; Pito 1996; Chen and Li 2005) use measurements to define a globally connected scene structure that is evaluated when proposing and selecting next best views.

Yuan (1995) presents a global scene representation based on *mass vector chains*. Point measurements are segmented into local surface patches such that the normal to each patch is consistent for all the encompassed points. The mass vector chain is the set of surface normals each weighted by the size of the corresponding surface patch. The paper shows that the sum of the mass vector chain for an enclosed convex surface is equal to zero. Virtual surface patches can be added and processed as part of the mass vector chain to handle concave surfaces. This approach does not use a set of view proposals. A next best view is selected with an orientation equal to the sum of the current mass vector chain. A view with this orientation should observe a surface patch, if one exists, whose normal is equal to the negative sum of the mass vector chain that would enclose the boundary surface. A method for setting the view distance is not discussed. The view orientation is adjusted if it is occluded by existing surface patches. A scene observation terminates when the sum of the mass vector chain is equal to zero (i.e, when an enclosed surface has been observed).

Pito (1996) presents a global *positional space* representation to encode information on the visibility of unobserved scene regions at the boundary of an observed scene mesh. The positional space is a discretised surface (e.g., a tessellated sphere) encompassing the scene. The scene volume is discretised into a voxel grid to denote unobserved regions. Sensor measurements are connected into a triangulated surface mesh. Rectangular *void patches* attached to each boundary edge in the mesh represent the transition between observed and unobserved space. The patch orientation is defined by identifying free space around the mesh using raycasting from the set of view proposals. These are sampled from a discretised surface encompassing the positional space. The intersection of *ranging rays* from a view proposal with a region of the positional space denotes its visibility. The same process is used to determine the observability of void patches from regions on the positional space along *observation rays* defined by their normals. The set of ranging rays which intersect with the same region of positional space as a set of observation rays denote that the void patches corresponding with the observation rays are visible from the views associated with the ranging rays. The next best view selection metric chooses the view that can observe the greatest number of void patches while also resampling a given proportion of known surfaces to ensure overlap between views. A scene observation terminates when the number of observable void patches falls below a specified threshold.

Chen and Li (2005) present a global *surface trend* scene representation. This requires that the scene surfaces are locally smooth and continuous. Sensor measurements are treated as noisy samples from a continuous globally connected surface defined by a geometric function. The difference between observed points and the surface trend is the residual error. A set of points is identified at the boundaries of the currently observed surface from which the scene observation can be extended into the unknown scene volume. The next best view target is determined by selecting the point from this set with the least uncertainty in the local surface trend (i.e., the lowest surface order) from which the greatest unknown scene volume can be observed. Boundary points on smoother surfaces are weighted higher, as they provide more

accurate predictions of the unseen surface geometry. The next best view pose is determined by considering the sensor parameters (i.e., the resolution and field-of-view) and the local surface geometry. The view orientation is set as the inverse of the surface normal to minimise the measurement uncertainty. The view position (i.e., the distance from the surface) is chosen to observe as much of the unknown surface as possible given the sensing constraints. Occlusion handling is discussed but no approach is included in the implementation. A termination criterion is presented based on scene coverage. The surface trend model is discretised and regions with no observed points (i.e., holes in the surface) are identified. The scene observation ends when no holes with a diameter greater than a specified threshold exist.

Approaches using global representations are often capable of obtaining high-quality observations of scenes with simple surface geometries, for which the global assumptions imposed on the scene structure are valid, but typically do not generalise to obtaining observations of scenes with complex and discontinuous surface geometry.

2.4 Volumetric Representations

A *volumetric* representation is most commonly used by NBV planning approaches for encoding scene information. This representation divides the scene volume into a three-dimensional grid of cells known as *voxels*. A state associated with each voxel encodes information on its observability, visibility and the measurements it contains. Views can be obtained from a view surface encompassing the scene, sampled using path planning techniques or proposed using voxel information. Occlusions and voxel visibility can be identified by raycasting the voxel grid. A next best view is usually selected by considering the states of the voxels visible from each view proposal. Termination criteria are often defined in terms of the observation states of voxels or as a fixed number of views.

Approaches using a volumetric representation can be broadly separated into three categories based on their method for proposing views. Many approaches sample a fixed set of views from a surface encompassing the scene (Connolly 1985; Wong et al. 1999; Papadopoulos-Orfanos and Schmitt 1997; Massios and Fisher

1998; Banta et al. 2000; Vasquez-Gomez et al. 2009; Vasquez-Gomez et al. 2014; Adler et al. 2014; Isler et al. 2016; Delmerico et al. 2018; Abduldayem et al. 2017; Mendoza et al. 2019). The view surface is usually a sphere or hemisphere depending on the sensing constraints. These approaches are discussed in Section 2.4.1.

Some approaches use path planning methods to sample view proposals from regions of free space in the scene (Potthast and Sukhatme 2011; Potthast and Sukhatme 2014; Yoder and Scherer 2016; Bircher et al. 2016; Bircher et al. 2018; Selin et al. 2019; Vasquez-Gomez et al. 2017; Vasquez-Gomez et al. 2018). These are discussed in Section 2.4.2. Other approaches propose views based on information obtained from point measurements (Connolly 1985; Wong et al. 1999; Daudelin and Campbell 2017; Monica and Aleotti 2018a). These are discussed in Section 2.4.3.

Connolly (1985) and Wong et al. (1999) present multiple view proposal techniques. Song and Jo (2017) also use a volumetric representation but this was extended to a combined approach in subsequent work (Song and Jo 2018) that will be discussed in its entirety in Section 2.6.

2.4.1 Sampling Views from a Surface

Volumetric approaches that sample views from a surface encompassing the scene primarily differentiate themselves by how they use information in their voxel representation to inform the selection of next best views and define termination criteria.

Connolly (1985) presents what is arguably the formative work for the field of NBV planning. This is the first work to use the term *next best view* and present approaches to the problem. One of these approaches uses a fixed set of view proposals sampled from the surface of a sphere encompassing the scene. Voxels are classified as *occupied* if they contain point measurements, *unobserved* if they have not yet been viewed or *unoccupied* if they have been observed but contain no point measurements. All voxels are initially classified as unobserved. A next best view is selected to observe the greatest number of visible unobserved voxels while accounting for occlusions from occupied voxels. A scene observation terminates when there are no remaining view proposals with visible unobserved voxels. A

similar approach is presented by Wong et al. (1999) as a benchmark for their work on information-based view proposals.

Papadopoulos-Orfanos and Schmitt (1997) present an approach using the same voxel state representation as Connolly (1985). A preplanned zigzag trajectory traverses views sampled uniformly within the scene volume. After each view is obtained the voxel representation is updated with newly classified occupied and unoccupied voxels. The trajectory is then modified to account for the obstacles and occlusions resulting from occupied voxels by moving into unoccupied voxel space. The scene observation terminates when the trajectory is complete.

Massios and Fisher (1998) introduce an observation quality metric to the voxel representation and a new class of *ocplane* voxels. This voxel class contains unobserved voxels with at least one neighbouring unoccupied voxel. The observation quality metric is based on the angle between a local estimate of the surface normal, computed from the point measurements within a given voxel neighbourhood, and the view orientation. This metric is updated for every occupied voxel observed by each view. The next best view selection metric chooses the view which maximises a weighted sum combining the number of observable ocplane voxels and the estimated improvement in observation quality for visible observed voxels. The termination criterion is triggered when a previously visited view is selected as the next best view.

Banta et al. (2000) present a set of next best view selection metrics that are integrated into a multistage observation approach. All of these metrics are based on the distribution of *occluded* voxels in the scene. Occluded voxels are unobserved voxels which lie within the sensor field-of-view but are obscured by occupied voxels. Initial measurements are obtained by greedily selecting next best views that can observe the greatest number of occluded voxels. Intermediate measurements are obtained by selecting the next best view that can best observe the centroid of occluded voxels. Subsequent measurements are obtained by clustering the occluded voxels and selecting the view with the best visibility of the largest cluster. A scene observation terminates when the ratio of observed to occluded voxels exceeds a given threshold and too few voxels have changed classification.

Vasquez-Gomez et al. (2009) present an approach, extended in subsequent work (Vasquez-Gomez et al. 2014), which combines the full set of voxel classifications (i.e., unobserved, occupied, unoccupied, occluded and occplane) with a next best view selection metric that considers observability, visibility, quality and navigation cost. The observability metric aims to balance the observation of unobserved voxels and obtaining overlapping views. It specifies target percentages for what proportion of the voxels visible from a view should be classified as occupied or occplane and penalises views which deviate from the targets. The visibility metric aims to identify occlusions. It evaluates views based on the number of visible occplane voxels relative to the raycasting resolution. The quality metric aims to improve measurement accuracy by prioritising views that are orthogonal to scene surfaces. An estimate of the local surface normal is computed for each occupied voxel. The metric is defined as the sum of the angles between the estimated normal for each voxel and the view orientation weighted by the number of occupied voxels. The navigation metric aims to reduce the distance travelled between views. The combined next best view selection metric is defined as the product of the visibility metric and the summation of the observability, quality and navigation metrics. This formulation prioritises extending the scene observation over other considerations. The termination criterion is described as ‘when the next best view does not provide new information’ but is not quantified.

Adler et al. (2013) present an approach that detects gaps (i.e., unobserved regions) in a scene observation using a particle simulation. The set of observed points is downsampled to obtain a given minimum interpoint distance and each point is assigned a radius. Particles of a given size are uniformly sampled above the scene. Their descent through the scene volume is modelled with a physics simulation. Particles which collide with an observed point and then continue to fall through the scene volume indicate the existence of holes. The observation value of a voxel is defined by the number of collisions between particles and observed points. This value is increased for every collision between a point in the voxel and a particle that successfully fell through the scene and whose final collision occurred in the voxel. A

trajectory is planned to observe the holes by placing a waypoint at the centre of each voxel with a nonzero value and ordering the waypoints by decreasing value.

Isler et al. (2016), in work later extended by Delmerico et al. (2018), present different next best view selection metrics for use with a probabilistic voxel representation. This approach and subsequently discussed works using probabilistic voxel representations commonly define next best view selection metrics using Information Gain (IG). IG metrics evaluate views based on the expected information (i.e., entropy value) contained within the set of voxels that are likely to be visible. In this work the entropy value of a voxel is determined by its occupancy likelihood (i.e., the probability that it contains point measurements). The IG associated with a view for each metric is determined from the visibility and entropy value of voxels.

The *occlusion aware* metric weights the occupancy value of each voxel by the probability of it being visible from a given view. The *unobserved voxel* metric restricts the *occlusion aware* evaluation to unobserved voxels. The *rear side voxel* metric counts the number of unobserved voxels which are occluded and neighbour occupied voxels. The *rear side entropy* metric weights the *rear side voxel* metric with the *occlusion aware* metric. The *proximity count* metric evaluates the distance of occluded voxels from the occupied voxel which obscures them up to a given maximum distance. A *combined* metric is defined as a weighted sum of the other metrics. A next best view is chosen to maximise the weighted difference between one of the IG metrics and the navigation cost of reaching the view relative to the cumulative IG and navigation cost for all views. The termination criterion is satisfied when the greatest IG associated with a view falls below a given threshold.

Abduldayem et al. (2017) present a multistage observation approach using a probabilistic voxel representation which leverages the symmetry of scenes to improve observations. An initial observation of the scene is obtained using a predefined trajectory planning method. Symmetry in the observation is detected and used to generate a set of predicted points. An entropy value is computed for each voxel from its occupancy likelihood. View proposals are sampled uniformly from unoccupied voxels in the scene volume. A next best view is selected to maximise an IG metric

that considers the cumulative entropy of all voxels visible from the view. A scene observation terminates when the estimated change in voxel entropy for the next best view remains lower than a given threshold for a specified number of views.

Mendoza et al. (2019) present a deep learning approach to NBV planning using a three-dimensional Convolutional Neural Network (CNN). A training dataset is generated by exhaustively searching the view space to determine the optimal sequence of next best views to completely observe a known scene. This sequence of views forms the classification set used for training the CNN. The CNN performance is assessed based on whether views are selected in the optimal order.

Approaches with a volumetric representation that sample views from a surface encompassing the scene are capable of obtaining observations with high coverage of the scene volume. However, the completeness of observations depends on the distribution of sampled views and high coverage of the scene volume does not necessarily ensure good surface coverage. Views are proposed without knowledge of the scene geometry and are not adjusted to account for occlusions.

2.4.2 Sampling Views with Path Planning

Approaches that use path planning techniques to sample views from the scene volume build a planning graph from the current sensor position into known free space and propose a view at each node which is evaluated with a next best view selection metric.

Potthast and Sukhatme (2011) and Potthast and Sukhatme (2014) present an approach which uses a probabilistic voxel representation and samples views for trajectories using the Probabilistic Road Map (PRM) planner (Kavraki et al. 1996). The occupancy probability of each voxel is updated by applying a stochastic sensor model, defined as a Gaussian probability density function, to sensor measurements. Voxels are classified as unobserved, occupied or unoccupied by thresholding their occupancy probability. Views are sampled randomly in the workspace of the sensor and checked for reachability constraints. The feasible set of views is connected into a PRM graph. A next best view is selected to observe the greatest number of unknown voxels. The best path from the current sensor pose to the next best

view is found by maximising the total number of unknown voxels visible from each view traversed in the graph.

Yoder and Scherer (2016) present an approach for observing large-scale scenes using an aerial platform. A *region of interest*, defined as a set of voxels that are expected to be fully observed, is specified within the scene volume. The existing occupied, unoccupied and unobserved voxel classes are used in addition to a *frontier* classification. A frontier voxel is defined as an unoccupied voxel which neighbours an occupied voxel and an unobserved voxel which are themselves neighbours. The IG for each view is evaluated based on the number of visible frontier voxels. Visibility is quantified by considering the distance of a frontier voxel from the view. Views are sampled around the scene with the SPARTAN path planner (Cover et al. 2013). The next best view is selected using a weighted sum of the IG and navigational cost. A scene observation terminates when no sampled views have a given minimum IG.

Bircher et al. (2016) and Bircher et al. (2018) present an approach that incrementally plans a trajectory to observe a scene using the RRT algorithm (LaValle 1998). A node in the RRT tree represents the position of a view proposal with an associated orientation that is also sampled. Views are proposed by growing an RRT tree from the current sensor position into unoccupied voxel space for a given number of view proposals or until a view proposal is found with a nonzero IG. The IG metric is defined by the cumulative volume of unobserved voxels visible from the views traversed in the tree between the current position and a given view. The next best view is selected to be the view with the highest IG and the first view along this path is obtained by the sensor. The voxel grid is then updated and a new RRT tree is grown. The scene observation terminates if no views with a nonzero IG are found after sampling a specified maximum number of view proposals.

Selin et al. (2019) present an approach which scales efficiently to observing large scenes by using a continuously valued IG function and sparse raycasting to evaluate view proposals sampled from free space in a scene with RRT. This work improves upon the approach presented by Bircher et al. (2016) and Bircher et al. (2018). The positions of view proposals are given by the RRT tree nodes. The best orientation for

each view is determined by discretising the space of potential orientations around the view and choosing the discrete orientation with the maximum IG. The continuously valued IG function is a Gaussian Process (GP) (Rasmussen and Williams 2006) defined using a set of frontier views sampled by RRT trees from previous iterations. The IG for a new view is given by the posterior mean of the GP if the posterior variance at that point is below a given threshold. If not, it is computed using a sparse raycasting method and the view is added to the set of frontiers defining the GP. The IG of a view is computed with sparse raycasting by identifying regions of unobserved space within the view frustum and computing their cumulative volume with cubature integration. A next best view is selected from a union of the frontier and currently sampled view sets to maximise the IG. The scene observation is terminated when no view proposal has an IG value above a given threshold.

Vasquez-Gomez et al. (2017) and Vasquez-Gomez et al. (2018) present approaches for planning views in the state space of a robotic platform. Vasquez-Gomez et al. (2017) use random sampling to obtain a set of view proposals while Vasquez-Gomez et al. (2018) present an RRT-based sampling method. The IG for each view is evaluated based on the amount of overlap with previous views, the number of visible unobserved voxels, the distance from the current robot state and whether the view is reachable. The next best view is chosen to maximise the product of these factors. In both cases a path to the next best view is planned using RRT. Termination criteria are presented based on a minimum number of visible unobserved voxels and whether there are any remaining reachable view proposals.

Volumetric approaches that sample view proposals using path planning techniques can typically obtain greater scene coverage than those that sample views from a surface encompassing the scene. Sampling views within unoccupied regions of the scene volume increases the likelihood that surfaces with restricted visibility from certain view orientations, often caused by self-occlusions, will be observable.

2.4.3 Proposing Views using Scene Information

Approaches that propose views using scene information can often obtain high-quality observations more efficiently than those that propose views with sampling-based methods. Observations are more efficient as fewer views need to be evaluated and higher quality as views are proposed using existing knowledge of the scene structure.

Connolly (1985) also presents an approach for proposing views based on scene information. Views are proposed and a next best view is selected based on the number of unobserved voxel faces oriented in each direction within the scene volume. Occlusions from occupied and other unobserved voxels are accounted for by only counting faces neighbouring unoccupied voxels. These counts represent the number of unobserved voxels visible from each side of the scene volume. Views are initialised to observe each corner of the scene volume so that multiple sides are visible simultaneously. The next best view is selected to have the greatest cumulative count of visible faces over its three adjacent sides. A similar approach is presented by Wong et al. (1999) but theirs does not account for occluding voxels.

Daudelin and Campbell (2017) present an approach that proposes views within a discretised bounding box encompassing part of the scene volume. This approach uses a probabilistic voxel representation and a frontier voxel classification. This classification differs from the one presented by Yoder and Scherer (2016) as frontiers are defined as unobserved voxels with both occupied and unoccupied neighbours. Next best views are selected using an IG metric that evaluates the cumulative value of observing every voxel in the scene from each view. The value of observing a target voxel from a given view is the product of its occupancy likelihood with the probabilities that it contains a scene surface and is visible from the view.

The scene surface probability (i.e., the probability that a voxel contains a scene surface) is equal to the occupancy likelihood for observed (i.e., occupied or known unoccupied) voxels. The scene surface probability for an unobserved voxel is defined by an exponential decay function based on its distance from the closest frontier voxel. The probability that a target voxel is visible from a given view is the product of one minus the scene surface probability for each voxel between the view and the target

voxel. The bounding box for proposing views is initially defined to encompass all occupied voxels in the scene and then expanded to include all nearby voxels within a given distance. The bounding box is discretised at a specified resolution and a view is proposed at the centre of each discrete cell. A next best view is chosen from the set of view proposals to maximise the IG metric. The scene observation terminates when the maximum IG associated with a view falls below a given threshold.

Monica and Aleotti (2018a) present an approach which identifies frontier voxels, as defined by Yoder and Scherer (2016), and proposes views to observe these voxels. The orientation of the view proposed to observe a frontier voxel is defined as the negative of a weighted average of the estimated normals for its neighbouring voxels. The view position is determined by moving a specified distance from the voxel centre in the direction of the estimated normal. An additional complement of views can also be sampled within a given solid angle of each view to ensure observation. Frontier voxels that are sufficiently close with similar normals are clustered together. The saliency of scene regions is computed by segmenting the point measurements and determining a saliency value for each segment. The segments are then ordered by decreasing saliency. The view proposals associated with the frontier voxels whose points are part of the most salient segment are selected. The next best view is selected as the view with the greatest number of visible frontier voxels. If no view exceeds a given threshold the view proposals from the next most salient segment are considered. The scene observation terminates after a fixed number of views.

Volumetric approaches that propose views using knowledge of the scene structure obtained from previous measurements can efficiently obtain high-quality observations. Computational efficiency is improved as it is only necessary to evaluate views that will improve the scene observation. More accurate measurements can be obtained by using geometric scene information to propose views with better surface visibility. The fidelity of the geometric information used is limited by the voxel resolution.

2.5 Surface Representations

NBV planning approaches using *surface* representations (Reed and Allen 2000; Hollinger et al. 2012; Khalfaoui et al. 2013; Roberts et al. 2017; Peng and Isler 2019) connect sensor measurements into a triangulated mesh that aims to approximate the scene geometry. This can provide high-fidelity information on the scene structure when proposing views, selecting a next best view and defining termination criteria based on observation quality. Occlusions are identified by raycasting the mesh. Mesh boundaries and measurement density can be used to consider scene coverage when proposing and selecting views or define quality-based termination criteria.

Most approaches using a surface representation (Reed and Allen 2000; Hollinger et al. 2012; Roberts et al. 2017; Peng and Isler 2019) utilise a multistage observation that requires an initial survey of the scene to be captured using a manual or preplanned trajectory. This can restrict their applicability to autonomous systems. Khalfaoui et al. (2013) do not use a multistage observation but state that their approach is limited to scenes with smooth and continuous geometry.

Kriegel et al. (2011) also present a surface-based approach but this was extended to a combined representation in subsequent work (Kriegel et al. 2012; Kriegel et al. 2015) that will be discussed in its entirety in Section 2.6.

Reed and Allen (2000) present an approach that uses a mesh to represent both *observed* and *occluded* surfaces. An initial model of the scene is obtained from a set of predefined views. The point measurements are connected into a triangulated mesh after each view is obtained. The mesh edges are then extruded along the view orientation until they intersect with the scene boundary. This produces a closed mesh consisting of *observed* surfaces represented by triangles whose vertices are all point measurements and *occluded* surfaces whose triangulated faces contain at least one vertex from the scene boundary. This representation is updated to incorporate new measurements after each view is obtained. The view planning objective is to improve scene coverage by observing the occluded surfaces. While most approaches discretely propose a set of independent views, this work computes a continuous *plan volume* which can then be discretised to obtain individual views.

The plan volume is defined by the intersections between *imaging*, *occlusion* and *placement* volumes. The imaging volume is the space from which the target (i.e., occluded) surfaces can be observed. The occlusion volume is the space from which the target surfaces are unobservable. The placement volume is defined by the reachability constraints of the sensor. The plan volume is given by the intersection of the placement volume with the set difference between the imaging volume and the occlusion volume. A set of views is obtained by discretising the plan volume. The next best view is chosen from this set to observe the greatest area of occluded surfaces. A fixed number of views is used for the experimental results presented but the use of a termination criterion based on a threshold of the number or area of remaining occluded surfaces is also discussed.

Hollinger et al. (2012) present an approach which models uncertainty in the surface representation of a scene and plans views to reduce this uncertainty. This approach requires that a coarse mesh reconstruction of the scene is obtained using a manual survey. Uncertainty in the mesh is modelled using *Gaussian process implicit surfaces*. The model considers both the density of points in the mesh and variability in the triangulated surface normals. Mesh regions with sparser measurements or a greater variation in surface normals have higher uncertainty. A set of views is proposed to observe the scene such that each point in the mesh is visible from a given minimum number of views. Two different next best view selection metrics are presented to reduce *coverage-based* uncertainty (i.e., increasing the mesh density) and *variance-based* uncertainty (i.e., reducing the variation in surface normals). These metrics are compared experimentally using a random view selection metric as a baseline. Termination criteria are presented based on a fixed number of views and a minimum threshold in uncertainty reduction.

Roberts et al. (2017) present an approach to trajectory optimisation for obtaining multiview stereo reconstructions of large-scale scenes. An initial scene observation is obtained by traversing a preplanned view trajectory to capture a set of images that is processed offline to compute a surface mesh. This initial observation determines a coarse estimate of the scene geometry to identify free space and inform

the computation of an optimised view trajectory. The set of view proposals is obtained by taking the Cartesian product between independently sampled sets of view positions and orientations. The view positions are sampled uniformly from the scene volume and the view orientations are uniformly sampled from a hemisphere encompassing the scene to point towards the scene centre. View proposals that do not lie in free space, as defined by the initial surface mesh, are rejected.

The value of each view proposal is quantified by assessing the visibility of a set of points uniformly sampled from the initial surface mesh. Each sampled point is encompassed by a hemisphere whose base lies on the triangulated mesh surface. A ray is cast from each view proposal to every sampled point. The point of intersection between a ray and the hemisphere of a sampled point defines a circular disk. The radius of this disk is defined by the view distance and the observation angle (i.e., the angle between the view orientation and the local surface plane). The value of each view proposal is given by the cumulative area of the circular disks for all points. The set of view proposals is subsampled before the trajectory optimisation stage by greedily selecting the best view orientation associated with each view position. An optimised trajectory is computed from this subset of view proposals to maximise the additive value of each view visited while satisfying a specified distance constraint.

Peng and Isler (2019) present a similar approach to obtaining large-scale multiview stereo reconstructions as Roberts et al. (2017). An initial scene observation is also required and obtained by traversing a preplanned view trajectory. Their key contribution is a method for proposing views that accounts for the scene geometry. View proposals are sampled from a manifold encompassing the scene. This is a smooth surface defined by the convex hull of the point set produced by moving each vertex in the initial mesh a given distance in the direction of its surface normal.

The coverage and quality of each face in the mesh is considered when sampling views from the manifold. Coverage is determined by the number of views within the *visibility cone* of a face. This is a cone of unit height with its apex at the centre of the face and an apex angle equal to the angle between the views from which the face was initially reconstructed. The quality of a face is defined as the coverage per unit

area of the face and weighted by the visibility angle to encourage the observation of poorly triangulated surfaces. View proposals are sampled from the manifold such that the coverage of each face exceeds a given threshold. The orientation of each view is chosen to maximise the observation of low quality faces. An optimal path for traversing the views is found by weighting the distance between views by the quality of faces visible from intermediate views along the straight line path. This prioritises paths that obtain greater coverage of poorly observed surfaces.

Khalifaoui et al. (2013) present an approach that does not require a multistage scene observation but is restricted to observing scenes with smooth surface geometry. The visibility of triangulated surfaces, referred to as *surface patches*, in the mesh representation is classified based on the angle between their surface normal and the view orientation. Surfaces whose visibility angle is greater than a given threshold are classified as *barely visible* while those with a sufficiently small angle are classified as *well visible*. The visibility of well visible surfaces is verified using raycasting. A set of potential views are proposed by applying *mean shift* clustering to the normals associated with the set of barely visible surfaces. The next best view is chosen to be the view in this set associated with the largest cluster of normals as the greatest number of barely visible surfaces should be observable. A minimum distance constraint between observed views is used to ensure their spatial distribution around the scene and define the termination criterion. A scene observation terminates when no remaining view proposals satisfy the distance constraint.

NBV planning approaches with surface representations are often capable of obtaining higher quality scene observations than those using volumetric representations as they consider the scene geometry when proposing and selecting views. However, the multistage observations used by most approaches require human intervention, precluding deployment on autonomous sensor platforms, or have restrictions on permissible scene geometry (e.g., Khalifaoui et al. 2013).

2.6 Combined Representations

NBV planning approaches using *combined* representations aim to leverage the advantages of multiple representations while mitigating their limitations. Most of these approaches combine surface and volumetric representations (Low and Lastra 2006; Krainin et al. 2011; Kriegel et al. 2012; Kriegel et al. 2015; Dierenbach et al. 2016; Song and Jo 2018) to utilise knowledge of the surface geometry and the observation state of scene volumes when proposing views, selecting a next best view and defining termination criteria. Monica and Aleotti (2018b) present a hybrid *surfel* representation which fuses concepts from surface and volumetric representations. Karaszewski et al. (2012) and Karaszewski et al. (2016b) combine structured and unstructured representations to perform multistage observations that acquire increasingly higher resolution scene models at each stage using different sensors.

Low and Lastra (2006) present a hierarchical approach which represents the scene using a combination of voxels and *surface patches*. The NBV planning objective is to obtain a given minimum density of point measurements within each occupied voxel. Surface patches are defined to represent points in occupied voxels with insufficient density. A surface patch is a bounding rectangle with a known surface area, an average point density and a density deficit equal to the difference between the minimum required density and the average density of the patch. Views are proposed from feasible *view volumes* represented with a voxel grid. The value of a view is determined by evaluating the sensor constraints and the estimated increase in measurement density for a target surface patch. The value of observing each patch from every view volume is evaluated. The hierarchical nature of the approach means that view-patch pairs with zero value can be divided into smaller view volumes or surface patches until a given minimum size is reached or a nonzero value is obtained. The next best view is selected to maximise the view value. A scene observation terminates when all occupied voxels satisfy the specified measurement density.

Krainin et al. (2011) present an approach that uses an underlying voxel representation to encode information on a maximum likelihood surface estimated from point measurements. Voxels are classified as unobserved, unoccupied or occupied if

they contain point measurements. View proposals are sampled uniformly from the surface of a sphere encompassing the scene with a minimum angular separation. A next best view is selected from the set of view proposals to provide the greatest predicted reduction in uncertainty for the estimated surface. A scene observation terminates when the predicted reduction in surface uncertainty falls below a given threshold for all remaining view proposals.

Kriegel et al. (2012) and Kriegel et al. (2015) extend earlier work using a surface representation (Kriegel et al. 2011) to present an approach using a combined representation. The surface-based approach proposes views to observe the boundaries of a triangulated mesh. Boundary edges are classified as left, right, top or bottom based on their orientation relative to the current sensor pose. A estimate of the boundary surface trend is computed for each edge set. Views are proposed to observe the unknown region beyond each boundary. Every view is oriented to observe a point sampled from the trend surface at a specified distance along the estimated normal. Points are sampled at a given step distance moving away from the observed mesh in a perpendicular direction to the boundary. New views are proposed until there is insufficient overlap between the observed mesh and sensor field-of-view. Next best views are selected from the set of view proposals in order of boundary class (i.e., left, right, top, bottom) and then increasing distance from the boundary. New views are proposed after every view which obtains a given minimum percentage of new measurements. The scene observation terminates when no views remain.

Kriegel et al. (2012) combine this surface-based approach with a probabilistic voxel representation. An IG metric for selecting next best views is presented which aims to maximise the cumulative entropy value of all voxels visible from a given view, as computed from their occupancy likelihood. A termination criterion is defined as a threshold on the percentage of boundary edges in the mesh. Kriegel et al. (2015) extend this work by selecting next best views using a weighted sum of the IG and a surface quality metric. The surface quality metric is defined as the sum of the boundary edge percentage and an average relative point density. The point density is computed voxelwise and specified as a relative proportion of a

given maximum density. It is weighted by the angle of incidence between the view orientation and the average normal of visible surfaces. An approach for assessing mesh coverage is presented to detect and fill holes in the mesh. Termination criteria are presented as thresholds on mesh coverage and the average relative point density.

Dierenbach et al. (2016) present an approach which learns a model of the scene surface from measurements using the Growing Neural Gas (GNG) algorithm (Fritzke 1994). The model is defined by a graph of connected nodes similar to a mesh representation. Every observed point is associated with the closest node. A Voronoi tessellation of the scene volume is computed from the model such that every node in the graph is the centroid of a Voronoi cell. The objective of this approach is to obtain a given minimum point density within each cell. Views are proposed to observe each node with an orientation equal to a surface normal estimate computed from the point measurements within their cell. A view is positioned at a given distance along the normal defined by the sensor parameters (i.e., resolution and field-of-view) and scene size. A next best view is selected to observe the Voronoi cell with the lowest density. The scene observation terminates when a given minimum point density has been obtained within each Voronoi cell.

Song and Jo (2018) present a combined approach which extends earlier work using a volumetric representation (Song and Jo 2017). The volumetric approach samples a set of view proposals using RRT* (Karaman and Frazzoli 2011). A next best view is selected from this set to maximise the number of visible unobserved voxels weighted by the view distance from the current sensor position, similar to the approach presented by Bircher et al. (2016) and Bircher et al. (2018). The shortest path to the next best view is planned using the RRT* tree. A new set of views is proposed within a given radius of the path. The number of frontier voxels, as defined by Daudelin and Campbell (2017), visible from each view is computed and each frontier is associated with the view that can observe it and the greatest number of other frontiers. Each view is weighted by its distance from the path. A coverage-sampling problem is solved to find the set of views with the smallest cumulative weight that can observe all of the frontiers. This

is repeated using smaller sampling radii until coverage of all the frontiers is no longer achievable. The shortest path to the next best view is computed which also visits these subviews. The scene observation terminates when the percentage of unobserved voxels falls below a given threshold.

Song and Jo (2018) combine their volumetric approach with a surface representation computed from point measurements as a maximum likelihood estimate, similar to the representation presented by Krainin et al. (2011). Points are sampled from the estimated surface and used to compute a Poisson reconstruction (Kazhdan et al. 2006). Each point is assigned a confidence value based on the average likelihood weight of its neighbours. Points with values lower than a given threshold are selected as targets for observation. These target points are clustered based on their distance and the angle between their surface normals, as defined by the Poisson reconstruction. Each cluster is represented by its average point. Views are sampled within a view frustum centred on each cluster and pointing in the direction of its normal. Views that are too far from the current path are excluded. Clusters with no feasible views are removed. The shortest path which traverses at least one view per cluster is computed. A set of subviews along this path are then found using the original approach (Song and Jo 2017). No new termination criteria is presented.

Monica and Aleotti (2018b) present a NBV planning approach using a surfel representation. Surfels are circular disks with a given position, radius and normal which are extracted from an underlying volumetric representation. Surfels which lie on the boundary between unobserved and unoccupied voxels are identified and classified as *frontels*. Next best views are selected to observe the set of visible frontels with the greatest surface area. View proposals are sampled from a sphere encompassing the scene. A termination criterion is defined based on the minimum observable frontel area.

Karaszewski et al. (2012) present a multistage observation approach for obtaining high resolution models of real world scenes in a controlled workspace. The approach combines a volumetric representation with a point classification based on point density. Initial observations of the scene are obtained using a preplanned view

trajectory until points are observed and at least one voxel is occupied. The NBV planning approach is then activated. At each iteration the set of point measurements is downsampled to obtain a given minimum interpoint distance. Points in this set with a point density greater than a specified minimum in a small radius and less than a given maximum density in a large radius are selected as targets for observation. A view is proposed to observe each point with an orientation defined by a normal vector computed for the point and a view position at a fixed distance from the point in the normal direction. A secondary NBV planning stage is used to observe sparse regions denoted by points with an insufficient density of measurements within a given radius. This radius is computed from the average interpoint distance for all measurements. Views are obtained until no sparse regions or reachable view proposals remain.

Karaszewski et al. (2016b) extend this approach to include a surface representation. The objective is to obtain an initial scene observation using a low resolution sensor with a large field-of-view which can then be refined using a high resolution sensor with a small field-of-view. The initial scene observation is performed using the original approach (Karaszewski et al. 2012). The refined observation is then performed based on a Poisson surface reconstruction of the initial observation. Points on the reconstructed surface are clustered based on their associated normals and a view is proposed to observe each cluster based on the average surface normal. The shortest path traversing this set of views is then computed and obtained.

Approaches that use combined representations can often obtain more complete and accurate scene observations than those that rely on a singular representation. This performance improvement typically incurs an increase in computational cost for maintaining multiple representations and the fusion of different metrics for selecting a next best view can require parameter tunings that do not generalise between scenes.

2.7 Discussion

The choice of a suitable approach and representation for obtaining scene observations typically depends on the availability of an *a priori* scene model, the structural complexity of the scene being observed, the desired resolution for an observation and

the constraints of the sensing system used. Model-based approaches are well suited to obtaining high-quality observations for the purpose of comparing a manufactured part with a known model but this requires *a priori* scene knowledge that is not available for unknown scenes. Model-free approaches address this limitation by proposing and selecting views without requiring any *a priori* scene information.

Approaches with global representations are often a suitable choice for obtaining high-quality observations of scenes with simple surface geometry. Their explicit consideration of the global connectivity between scene surfaces makes it possible to obtain complete scene observations without gaps in the observed surface geometry. However, the global assumptions imposed on the scene structure typically prevent these approaches from observing scenes with discontinuous surface geometry.

Approaches with volumetric representations are well suited to observing scenes for the purpose of determining the occupancy of scene volumes. They are capable of obtaining high-resolution observations of small-scale scenes when it is possible to represent the scene volume with a dense voxel grid (i.e., a small voxel size). However, due to the computational complexity of evaluating a voxel representation they typically do not scale to obtaining high-resolution observations of large-scale scenes as it is necessary to use a sparser voxel grid (i.e., a larger voxel size and lower resolution) in order to maintain a reasonable computation time.

Approaches with surface representations are often a good choice for obtaining high-quality observations of large-scale scenes as they can utilise high-fidelity information on the scene structure that is encoded in a triangulated mesh. However, in most cases it is prohibitively expensive to compute the triangulation of a mesh between dense sensor measurements online in real-time. To mitigate this cost many surface approaches utilise a multistage observation. Sparse measurements of the scene surfaces are first obtained by performing a manual survey and used to compute a rough triangulated mesh offline. The views for a secondary observation are then proposed based on this mesh to obtain dense sensor measurements. As the initial manual scene survey typically requires the oversight of a human operator it is often not possible to deploy surface approaches on an autonomous sensing system.

Approaches with combined representations aim to leverage the strengths of different representations while mitigating their limitations. Approaches using both volumetric and surface representations (e.g., Kriegel et al. 2015; Song and Jo 2018) can propose and select views that obtain good coverage of both the scene volume and surface structures. This improvement in observation performance comes with an increase in computational cost from maintaining and evaluating multiple representations. Combined approaches often do not generalise easily between scenes as determining a suitable combination of different next best view selection metrics and termination criteria introduces additional complexity in their parametrisation.

The unifying trait of the approaches discussed in this literature review is that they all use *structured* scene representations. The high-level organisation of point measurements provided by imposing an external structure on the scene makes it possible to clearly define the coverage of an observation and allows the visibility of unobserved regions and surface boundaries to be evaluated using raycasting. The assumptions made by structured representations typically aim to simplify the problem of considering scene coverage when proposing and selecting next best views at the expense of reducing the fidelity of the scene information represented and incurring a computational cost to maintain an external scene structure.

Global representations consider all sensor measurements to be part of a globally connected scene structure. This relies on an assumption that the scene being observed consists of surfaces with continuous geometry. When observing scenes with discontinuous geometry for which this assumption is invalid it may not be possible to obtain a complete observation.

Volumetric representations approximate the observation state of scene regions by considering the occupancy of point measurements within voxels. The accuracy of the encoded observation states is determined by the voxel resolution. Using smaller voxels (i.e., a higher resolution) increases the fidelity of the occupancy information but also increases the number of voxels. This means a greater computational cost is incurred when raycasting the voxel grid, as discussed by Low and Lastra (2006) and Monica and Aleotti (2018b).

Raycasting is performed to update the observation state of voxels with new measurements and evaluate the visibility of voxels from proposed views when selecting a next best view. The cost of raycasting a voxel grid increases cubically with the voxel resolution (i.e., as the voxel size decreases) and linearly with the number of views that require raycasting. This cost is partially mitigated by volumetric approaches that sample views using path planning techniques or propose views based on scene information as a smaller set of view proposals needs to be evaluated when selecting a next best view. Work has also been presented on improving the efficiency of raycasting for volumetric representations by Low and Lastra (2006), Vasquez-Gomez et al. (2013), and Selin et al. (2019) but such techniques are not yet widely adopted.

Surface representations aim to approximate the geometry of observed surfaces by inferring the connectivity of a triangulated mesh between point measurements. The validity of approximated surfaces depends on the structural assumptions made when connecting the mesh and the accuracy of the sensor measurements. They can deviate significantly from the underlying scene structure when using invalid connectivity assumptions or measurements from a noisy sensor. This can negatively impact the proposal and selection of views as the mesh boundaries may not denote the true coverage of a scene observation. Erroneous surface triangulations can produce invalid results when detecting occlusions and evaluating scene visibility.

Mesh computation is often performed offline due to the computational cost of obtaining a triangulated mesh from dense sensor measurements in real-time, as discussed by Peng and Isler (2019). Some combined approaches with surface representations compute a mesh online but these use a downsampled set of points to reduce the cost (e.g., Song and Jo 2018) or techniques for incremental mesh construction (e.g., Kriegel et al. 2011 use the method presented by Bodenmüller 2009).

Combined representations typically represent scene observations using both a voxel grid and a triangulated surface mesh. An increased computational cost is incurred by maintaining both representations but it is possible to mitigate the structural assumptions imposed by each individual representation. This is often

achieved by proposing and selecting views that can extend coverage of the surface mesh while also observing voxels whose visibility was occluded from previous views.

Many of the limitations associated with using structured scene representations can be overcome by using an *unstructured* representation. In this type of representation information on the state of a scene observation is directly encoded by the point measurements rather than being aggregated into an external structure. Fewer assumptions are made about the scene geometry and a lower computational cost is incurred as it is not necessary to maintain or raycast an external structure.

This thesis presents work on NBV planning with a novel unstructured *density* representation. This representation is based on the idea that a sufficient condition for obtaining complete scene observations is capturing a given minimum density of measurements from visible scene surfaces. All information on an observation is represented with a pointwise encoding and only point-based computations are performed when maintaining and evaluating the representation. This means that the computational cost of the representation scales with the number of observed points rather than the size of an externally imposed structure and its associated raycasting cost. Measurements are classified based on the local density of observed points rather than being aggregated into an external structure so the fidelity of scene information represented is not constrained by any structural resolution.

The proceeding chapter presents a NBV planning approach that uses this novel representation. The density of measurements in the scene is represented by classifying each observed point as a *core*, *frontier* or *outlier* based on the number of neighbouring points with a given radius. A view is proposed to observe each frontier point with an orientation defined by a normal estimated from local measurements and a position at a given distance from the point in the normal direction. If point measurements can not be obtained in the neighbourhood of a frontier point (i.e., the view is occluded) then the view can be adjusted until the frontier becomes visible. A next best view is selected to observe a frontier point close to the current sensor position while reducing the distance moved away from the initial view of the scene. An observation terminates when all frontier points are successfully observed.

3

Planning Next Best Views with an Unstructured Representation

Contents

3.1	Existing Methods	46
3.2	The Surface Edge Explorer (SEE)	47
3.2.1	Point Classification	49
3.2.2	Surface Geometry Estimation	52
3.2.3	View Proposals	54
3.2.4	Next Best View Selection	56
3.2.5	View Adjustment	57
3.2.6	Completion	61
3.3	Evaluation	61
3.3.1	Simulated Sensors	65
3.3.2	View Constraints	65
3.3.3	Algorithm Parameters	66
3.3.4	Performance Metrics	67
3.4	Discussion	67

This chapter presents the Surface Edge Explorer (SEE), a NBV planning approach with a novel unstructured density representation. Information on the state of a scene observation is directly encoded in the point measurements using a density-based classification. Observed points are classified based on the number of neighbouring measurements with a given resolution radius. This classification is used to define a *frontier* between surfaces in the scene that are fully and partially

observed. Sensor views are proposed to observe this frontier and expand the fully observed surfaces. Views are selected and new measurements are obtained until the entire scene is observed at the chosen resolution and measurement density. The representation is computationally efficient to maintain as only local updates are performed when new measurements are obtained. This enables SEE to obtain highly complete observations of scenes at any scale (i.e., from *bunnies* to *buildings*).

The work in this chapter was first presented in Border et al. (2017) at the 2017 Joint Industry and Robotics CDTs Symposium and extended in Border et al. (2018) at the 2018 IEEE International Conference on Robotics and Automation.

The experimental results presented in this chapter correct a mistake in Border et al. (2018). The implementation of volumetric approaches which produced the results in Border et al. (2018) used an erroneous sampling procedure that resulted in a nonuniform distribution of views proposals being sampled from the view surface. Experimental results presented in this chapter and throughout this thesis use a corrected implementation that samples a uniform distribution of view proposals.

The remainder of this chapter is organised as follows. Section 3.1 presents a review of existing NBV planning approaches that consider measurement density. The methodology of SEE is presented in Section 3.2. The classification of point measurements that defines a pointwise frontier between completely and partially observed scene surfaces is discussed in Section 3.2.1. The process of estimating the local surface geometry around frontier points is detailed in Section 3.2.2. The use of these local surface geometry estimates for proposing views to improve the coverage of a scene observation is explained in Section 3.2.3. The next best view selection metric used by SEE is presented in Section 3.2.4. The technique used to apply view adjustments when a target frontier point is not successfully observed from its associated view is described in Section 3.2.5. The density-based termination criterion used is presented in Section 3.2.6. An experimental comparison of the observation performance for SEE and state-of-the-art volumetric NBV planning approaches on four standard models and a full-scale building model is presented in Section 3.3. The results are discussed in Section 3.4.

The work presented in this chapter makes five key contributions:

1. A novel unstructured scene representation based on measurement density.
2. A method for proposing views directly from point-based scene information.
3. A novel technique for reactively handling occlusions that is capable of successfully observing target surfaces by using incrementally adjusted views.
4. A termination criterion that considers observation completeness by ensuring that a minimum measurement density is obtained for all observable surfaces.
5. Experimental results demonstrating that SEE is capable of obtaining scene observations with similar or better coverage than state-of-the-art volumetric approaches using fewer views and a significantly lower computation time.

3.1 Existing Methods

Approaches to NBV planning using structured representations have been presented that consider measurement density (e.g., Low and Lastra 2006; Kriegel et al. 2015; Dierenbach et al. 2016; Karaszewski et al. 2016b) but to the best of my knowledge this is the first work on NBV planning to use a purely unstructured representation that classifies the observation state of every point measurement based on the local density of neighbouring measurements and uses this pointwise classification to propose views, select a next best view and define a completion criterion.

The approach presented by Low and Lastra (2006) considers the average measurement density on surface patches extracted from voxels. Dierenbach et al. (2016) aim to obtain a minimum measurement density within Voronoi cells that segment the scene volume. Kriegel et al. (2015) consider the average point density within voxels when selecting next best views using a surface quality metric and defining termination criteria. Karaszewski et al. (2016b) identify a set of boundary points with low neighbourhood densities from a subset of point measurements and propose views to observe these points. The radius and density are computed from the sensor parameters and existing measurements rather than being user-specified.

The consideration of measurement density in SEE differs from these approaches as it is used to define an unstructured representation that does not require any

a priori assumptions about the scene structure. All knowledge of the scene is computed pointwise so the fidelity of encoded information (e.g., measurement density) is not limited by the size of voxels or the connectivity of a surface mesh. This allows SEE to propose and select next best views that can improve a scene observation until a given measurement density is achieved for all observable surfaces.

3.2 The Surface Edge Explorer (SEE)

SEE aims to obtain scene observations with a specified minimum measurement density over all observed surfaces (Sec. 3.2.1). This measurement density is defined by a resolution radius, r and target density, ρ , used to detect frontiers in the measurements. The resolution radius should be sufficiently large to robustly handle measurement noise while still being small enough to maintain computational efficiency. The target density is chosen to be sufficiently large to classify points at the specified resolution and attain a desired level of structural detail from the scene.

Frontiers are detected by classifying point measurements based on the number of neighbouring points within the resolution radius. Points with sufficient neighbours (i.e., the local density is greater than or equal to ρ) are classified as *core* points and those without are classified as *outlier* points. Outlier points with both core and outlier neighbours are then classified as *frontier* points. These frontier points represent the boundary between fully and partially observed surfaces.

Observation coverage is improved by obtaining measurements of the scene surfaces around frontier points (Sec. 3.2.2). Views are proposed to observe the frontiers by estimating the local surface geometry from points in their r -radius neighbourhoods. The surface geometry is described by a set of orthogonal vectors computed from an eigendecomposition of the neighbouring points. These are used to represent the local surface normal, a boundary between the fully and partially observed surface regions and the direction of partial observation (i.e., the frontier).

A view is proposed to observe the local surface region around each frontier point (Sec. 3.2.3). The view orientation is given by the estimated surface normal so that it is orthogonal to the locally estimated surface geometry. This orientation

is chosen to maximise surface coverage and improve measurement accuracy. The view position is placed at a given view distance, d , along the surface normal. A suitable distance can be specified empirically by the user or computed analytically from the algorithm parameters and sensor properties, as discussed in Section 6.1.2.

Next best views are selected from the set of *view proposals* using a metric which aims to reduce the sensor travel distance from both the current sensor position and the initial view (Sec. 3.2.4). The objective is to penalise the selection of a series of views that obtain measurements in a single direction without first capturing views of partially observed surfaces that are proximate to the initial sensor position.

When a frontier point lies near a surface discontinuity (e.g., an edge or corner) it is often not possible for a view with an orientation orthogonal to the surface on one side of the discontinuity to observe the opposing side (Sec. 3.2.5). This can render a successful observation of the frontier unobtainable as it is not possible to extend the scene observation beyond the surface discontinuity. It is possible to obtain a successful observation by iteratively adjusting unsuccessful views based on the measurements obtained until the frontier point is observed or a sufficient number of attempts have been made to classify it as an outlier. Points classified as outliers are not reprocessed unless new measurements are obtained within their r -radius.

Next best views are selected until no frontier points remain and all measurements have been classified as core or outlier points (Sec. 3.2.6). The extent of a scene observation can be bounded by discarding all points outside of a given volume.

When obtaining some observations, in particular those using high-resolution sensors or of large-scale scenes, it is necessary to enforce a minimum separation, ϵ , between point measurements in order to maintain an upper bound on memory consumption and computational cost. The size of this separation is typically chosen based on the target density. In this case, new measurements are only added to the pointcloud observation if their ϵ -radius neighbourhood contains no existing points.

An overview of SEE is shown in Algorithm 1. Sensor measurements are obtained and processed until there are no remaining frontier points (Line 4). A set of measurements, M , is obtained from the current view, $\mathbf{v}$ (Line 5). The measurements

Algorithm 1 SEE($\mathbf{v}_0, r, \rho, d$)

```

1:  $\mathbf{v} \leftarrow \mathbf{v}_0$                                  $\triangleright \mathbf{v}$  is the current view and  $\mathbf{v}_0$  is the initial view
2:  $\mathbf{f} \leftarrow \text{NULL}$                            $\triangleright \mathbf{f}$  is the target frontier point
3:  $P = C = F = O = \emptyset$                      $\triangleright P$  is the complete point set,  $C$  is the core set,
                                                 $F$  is the frontier set and  $O$  is the outlier set

4: while  $F \neq \emptyset$  or  $\mathbf{f} = \text{NULL}$  do
5:    $M \leftarrow \text{GET-MEASUREMENTS}(\mathbf{v})$ 
6:    $P, C, F, O \leftarrow \text{CLASSIFY-MEASUREMENTS}(M, C, F, O, r, \rho)$ 
7:   if  $F \neq \emptyset$  then
8:     if  $\mathbf{f} \in F$  then
9:        $\mathbf{v} \leftarrow \text{ADJUST-VIEW}(M, \mathbf{v}, \mathbf{f})$ 
10:    else
11:       $Y \leftarrow \text{ESTIMATE-GEOMETRY}(P, F, \mathbf{v}, r)$ 
12:       $W \leftarrow \text{GET-VIEW-PROPOSALS}(Y, F, d)$ 
13:       $\mathbf{v}, \mathbf{f} \leftarrow \text{SELECT-NEXT-BEST-VIEW}(W, \mathbf{v}, r)$ 
14: return COMPLETE

```

are added to the pointcloud observation, P , and the classifications of core, C , frontier, F , and outlier, O , points are updated (Line 6). If any frontier points remain after the classifications are updated then a new view is chosen (Line 7). If the target frontier point, $\mathbf{f}$, associated with the current view was not successfully observed (i.e., it is still classified as a frontier point) then the view is adjusted (Line 8–9). Otherwise a new view is selected (Line 10). The local surface geometry around frontier points is estimated from neighbouring points within an r -radius (Line 11). Views are proposed to observe the frontiers based on the estimated surface geometry, Y (Line 12). A next best view is then selected from the set of view proposals, W , to improve the scene observation around a target frontier point (Line 13). The scene observation is considered complete when there are no frontier points remaining (Line 14).

3.2.1 Point Classification

Frontiers between fully and partially observed scene surfaces are identified by performing classifications of point measurements based on the local measurement density. Points are classified as either a core, frontier or outlier point based on the number of neighbouring points, k , with a radius, r , of the point (Fig. 3.1). The number of observed points in the r -radius is compared with the minimum number of points, $k_{\min}$, necessary to satisfy the desired point density, ρ , where $k_{\min} = \frac{4}{3}\rho\pi r^3$.

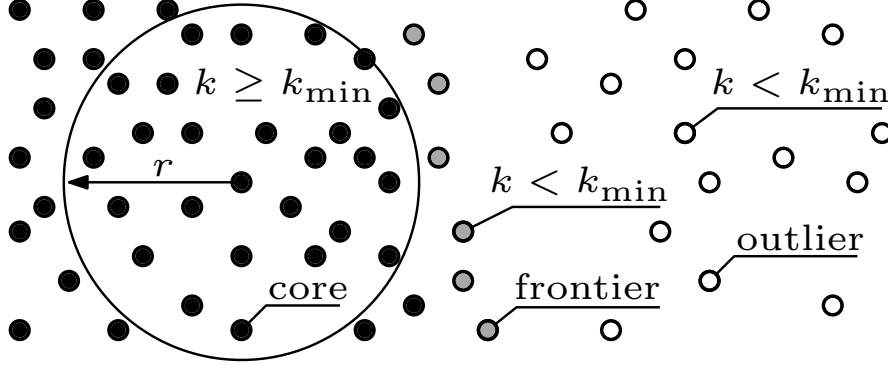


Figure 3.1: An illustration of the density-based classification approach used by SEE. Points with a sufficient number of neighbours, $k_{\min}$, in an r -radius are classified as core points (black) while those without are outlier points (white). Points with both core points and outlier points in their neighbourhood are frontier points (grey).

The approach used for density-based classification is based on the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm present by Ester et al. (1996). Point measurements, $P := \{\mathbf{p}_i\}_{i=1}^n$ where $\mathbf{p}_i \in \mathbb{R}^3$, are classified as core points, C , frontier points, F , or outlier points, O . The point classifications are complete and unique (i.e., every point is assigned to a single class) such that

$$P = C \cup F \cup O \quad \text{and} \quad C \cap F = C \cap O = F \cap O = \emptyset. \quad (3.1)$$

The set of points, $N_{\mathbf{p}}$, in the pointcloud within an r -radius of a point, $\mathbf{p}$, is given by

$$N_{\mathbf{p}} := N(P, r, \mathbf{p}) := \{\mathbf{q} \in P \mid \|\mathbf{q} - \mathbf{p}\| \leq r\}, \quad (3.2)$$

where $\|\cdot\|$ denotes the L^2 -norm.

A point is classified as a core point if it has more than $k_{\min}$ neighbours,

$$C := \{\mathbf{p} \in P \mid |N_{\mathbf{p}}| \geq k_{\min}\}, \quad (3.3)$$

where $|\cdot|$ denotes the cardinality of a set.

It is classified as a frontier point if it has both core and outlier neighbours,

$$F := \{\mathbf{p} \in P \mid |N_{\mathbf{p}}| < k_{\min} \wedge N_{\mathbf{p}} \cap C \neq \emptyset \wedge N_{\mathbf{p}} \cap O \neq \emptyset\}. \quad (3.4)$$

Remaining points that are not classified as core or frontier points are outliers,

$$O = P \setminus (C \cup F). \quad (3.5)$$

Algorithm 2 CLASSIFY-MEASUREMENTS(M, C, F, O, r, ρ)

```

1:  $V \leftarrow \emptyset$   $\triangleright V$  is the set of measurements that have been processed
2:  $k_{\min} \leftarrow \frac{4}{3}\rho\pi r^3$   $\triangleright k_{\min}$  is the number of neighbours required for a core point
3:  $P := C \cup F \cup O \cup M$   $\triangleright P$  is the complete point set,  $C$  is the core set,  $F$  is the frontier set,  $O$  is the outlier set and  $M$  is the new point set
4: for all  $\mathbf{p} \in M$  do
5:   if  $\mathbf{p} \notin V$  then
6:      $Q \leftarrow N(P, r, \mathbf{p}) \cup \{\mathbf{p}\}$ 
7:     for all  $\mathbf{q} \in Q$  do
8:       if  $\mathbf{q} \notin C$  then
9:          $N_{\mathbf{q}} \leftarrow N(P, r, \mathbf{q})$ 
10:        if  $|N_{\mathbf{q}}| < k_{\min}$  then
11:          if  $N_{\mathbf{q}} \cap C \neq \emptyset$  and  $N_{\mathbf{q}} \cap O \neq \emptyset$  then
12:             $F \leftarrow F \cup \{\mathbf{q}\}$ 
13:            if  $\mathbf{q} \in O$  then  $O \leftarrow O \setminus \{\mathbf{q}\}$ 
14:          else
15:             $O \leftarrow O \cup \{\mathbf{q}\}$ 
16:        else
17:           $C \leftarrow C \cup \{\mathbf{q}\}$ 
18:          if  $\mathbf{q} \in F$  then  $F \leftarrow F \setminus \{\mathbf{q}\}$ 
19:          if  $\mathbf{q} \in O$  then  $O \leftarrow O \setminus \{\mathbf{q}\}$ 
20:          if  $\mathbf{q} \neq \mathbf{p}$  and  $\mathbf{q} \notin V$  then
21:             $Q \leftarrow Q \cup N_{\mathbf{q}}$ 
22:             $V \leftarrow V \cup \{\mathbf{q}\}$ 
23: return  $P, C, F, O$ 

```

The procedure used for classifying the point measurements obtained from a view is shown in Algorithm 2. When a view is obtained the set of new measurements, M , is combined with the existing classification sets, C , F and O (Line 3). Each new point, $\mathbf{p} \in M$, is processed and added to either the core, frontier or outlier point sets (Line 4). Any new point that has not yet been processed is added to the (re)classification queue, Q , along with its neighbourhood points (Lines 5–6). If a point in the queue is not a core point then it is (re)classified based on the new measurements (Lines 7–9). Points with insufficient neighbours to be core are classified as frontier points if they have both core and outlier neighbours or otherwise as outlier points (Lines 10–15). Points with sufficient neighbours are classified as core points (Lines 16–19). If a point has not yet been processed and it is (re)classified as a core point then its neighbourhood is added to the (re)classification queue and it is marked as processed (Lines 20–22).

Classifying observed points based on the local measurement density distinguishes scene regions that have a sufficient point density to be considered completely observed (i.e., consist of core points) from those that require additional observations (i.e., contain frontier and outlier points). Distinct classifications for frontier and outlier points can be used to differentiate between sparse measurements from true surfaces and points that are the product of sensor noise. Measurements that are farther from previously obtained views are more likely to be the product of sensor noise. Frontiers are therefore identified at the boundary between regions of core and outlier points. These frontier points are used to propose views that will improve the observation of a scene by considering the local surface geometry around each frontier.

3.2.2 Surface Geometry Estimation

A scene observation is improved by obtaining new measurements around frontier points that can increase the local point density and expand the boundary between partially and completely observed surfaces. Views from which suitable measurements can be obtained are identified by considering the local surface geometry and the distribution of measurements around frontier points. A planar estimate of the local geometry is used to define a surface normal. This estimate is used in conjunction with the distribution of measurements in the frontier point neighbourhood to identify a frontier vector that points towards the region of partial observation and a boundary vector that points along the border between partially and fully observed surfaces.

A planar estimate of the local geometry around a frontier point, $\mathbf{f}$, is computed via an eigendecomposition of a matrix representation of its point neighbourhood,

$$\mathbf{D} := [\mathbf{p}_1 - \mathbf{f}, \dots, \mathbf{p}_n - \mathbf{f}] \in \mathbb{R}^{3 \times |N_f|}, \quad (3.6)$$

where $\mathbf{p}_i \in N_f$ are points from the r -radius neighbourhood computed using (3.2).

The decomposition of the covariance matrix, $\mathbf{A} := \mathbf{D}\mathbf{D}^T$, produces eigenvalues, $\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ and eigenvectors, $\Upsilon = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, satisfying the eigenequation,

$$\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i, \quad i = \{1, 2, 3\}. \quad (3.7)$$

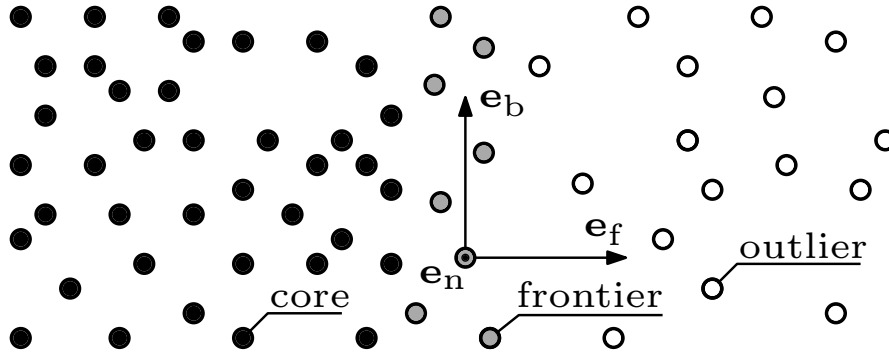


Figure 3.2: An illustration of the local surface geometry estimate defined by an orthogonal set of vectors. These vectors are orientated normal to the estimated surface, $\mathbf{e}_n$ (out of the page), point towards the region of partial observation, $\mathbf{e}_f$, and lie along the boundary between fully and partially observed scene regions, $\mathbf{e}_b$.

The matrix $\mathbf{A}$ is real and orthogonal. This means that the set of eigenvectors form an orthonormal basis (i.e., three mutually orthogonal unit vectors) of $\mathbf{D}$. Each eigenvector describes one component of the observed surface geometry (Fig. 3.2). The normal vector, $\mathbf{e}_n$, is orthogonal to the surface plane. The frontier vector, $\mathbf{e}_f$, lies in the surface plane and points in the direction of partial observation. The boundary vector, $\mathbf{e}_b$, points along the border between partially and fully observed surfaces.

The assignment of eigenvectors as surface geometry components is computed based on their eigenvalues, the angle between a vector and the observing view orientation, ϕ_o , and the mean point for the r -radius neighbourhood of the frontier.

The normal vector, $\mathbf{e}_n$, points along the axis associated with the least variance in neighbouring points. It is assigned to the eigenvector with the minimum eigenvalue,

$$\mathbf{e}_n = \{\mathbf{v}_i \mid \lambda_i = \min \{\Lambda\}\}, \quad (3.8)$$

and given a sign to point in the opposite direction to the observing view orientation,

$$\mathbf{e}_n \cdot \phi_o < 0. \quad (3.9)$$

The frontier vector, $\mathbf{e}_f$, points towards the partially observed region of the scene and is identified by considering the mean of the point neighbourhood for the frontier,

$$\bar{\mathbf{p}} = \frac{1}{|N_f|} \sum_{\mathbf{p} \in N_f} (\mathbf{p} - \mathbf{f}). \quad (3.10)$$

It is an unassigned eigenvector that has the greatest dot product with the mean point,

$$\mathbf{e}_f = \arg \max_{\mathbf{v} \in \mathcal{T} \setminus \mathbf{e}_n} (\bar{\mathbf{p}} \cdot \mathbf{v}), \quad (3.11)$$

and points in the opposite direction to the vector from the frontier point to the mean point, such that it points into the partially observed region of the scene,

$$\mathbf{e}_f \cdot \bar{\mathbf{p}} < 0. \quad (3.12)$$

The boundary vector, $\mathbf{e}_b$, is locally tangential to the border between the density regions and is assigned to the remaining eigenvector. The direction of the boundary vector is given by the cross product of the normal and frontier vectors,

$$\mathbf{e}_b := \mathbf{e}_n \times \mathbf{e}_f. \quad (3.13)$$

These orthogonal vectors represent the estimated surface geometry and distribution of neighbouring measurements around a frontier point. They are used to inform the proposal and adjustment of views. The initial view of a frontier point is proposed using the normal vector in order to maximise coverage of the surrounding scene surfaces. An adjustment of the view is computed using the frontier and boundary vectors if the initial view does not successfully observe the frontier point.

3.2.3 View Proposals

The coverage and accuracy of the measurements obtained from a view depends on the distance and angle of the sensor pose relative to the observed scene geometry. A view, $\mathbf{v} = \{\mathbf{x}, \phi\}$, is defined by a position, $\mathbf{x}$, and an orientation, ϕ . Views that are farther away from the scene surface can typically obtain greater coverage as a larger region of the scene is visible within the viewing frustum of the sensor. However, the measurement accuracy of many depth sensors degrades with distance (e.g., quadratically for stereo cameras) and therefore it is often desirable to use shorter view distances. The surface coverage and measurement accuracy obtained from a given view position is greatest for most sensors when the view orientation is orthogonal to the surface being observed. This provides the largest area of intersection between

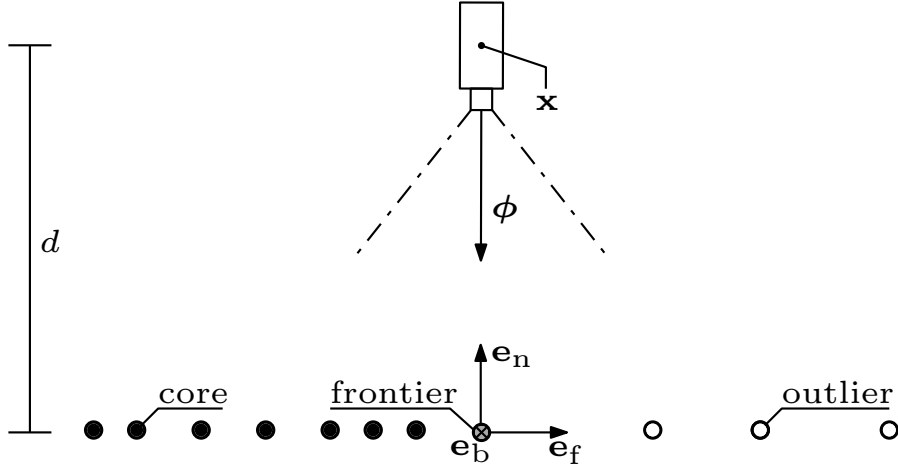


Figure 3.3: An illustration of the method for proposing views. A view, $\mathbf{v} = \{\mathbf{x}, \phi\}$, is proposed using the estimated local surface geometry, $\mathbf{e}_n$, $\mathbf{e}_f$ and $\mathbf{e}_b$ to observe the scene surfaces around each frontier point (grey). The view orientation, ϕ , is given by the inverse sign of the normal vector, $\phi = -\mathbf{e}_n$. The view position, $\mathbf{x}$, is set at a given view distance, d , from the frontier point along the normal vector, $\mathbf{e}_n$.

the viewing frustum and scene surfaces within the view distance when the scene geometry is locally planar. Sensors that compute depth measurements using feature triangulation (e.g., stereo cameras) can obtain greater measurement accuracy when a view is orthogonal to the surface as matching features can be identified more robustly.

Views are proposed to improve the observation of a scene by obtaining measurements from the surfaces around frontier points. A view is proposed to observe the scene around each frontier point based on the local surface geometry (Fig. 3.3).

Each view, $\mathbf{v} = \{\mathbf{x}, \phi\}$, is positioned at a given view distance, d , from its associated frontier point, $\mathbf{f}$, along the estimated normal vector, $\mathbf{e}_n$,

$$\mathbf{x} = \mathbf{f} + d\mathbf{e}_n, \quad (3.14)$$

with a view orientation given by the inverse sign of the normal vector,

$$\phi = -\mathbf{e}_n. \quad (3.15)$$

The proposed views are centered on their associated frontier points rather than being offset in the direction of partial observation (i.e., along the frontier vector) to ensure that the density of sensor measurements obtained from a view is greatest within the r -radius neighbourhood of its corresponding frontier point. When a

proposed view is offset from its target frontier then the overlap between this r -radius neighbourhood and the distribution of points within the sensor frustum decreases. This effect is particularly significant for sensors with a narrow field-of-view. Not using an offset therefore mitigates the risk of insufficient measurements being obtained in the neighbourhood of a frontier point for it to be successfully observed.

Sensor measurements captured from the proposed views can often successfully observe frontier points and improve the scene observation when a good estimate of the local surface geometry is obtained. Next best views are selected from the set of view proposals to try and improve coverage of the scene around nearby frontier points.

3.2.4 Next Best View Selection

Next best views are selected to improve the observation of a scene. The most efficient observations can be obtained by considering scene coverage and observation cost when selecting views. The challenge of assessing scene coverage with an unstructured representation is investigated in Chapter 5. The view selection metric presented in this chapter is formulated to reduce the observation cost, as quantified by the sensor travel distance required to obtain a scene observation. This quantification is used as the distance travelled between views when observing a scene determines the operation time and energy consumption utilised by a sensor platform (e.g., a drone). The view selection metric selects a next best view from the set of view proposals,

$$W := \{\mathbf{g}(\mathbf{f} \in F)\}, \quad (3.16)$$

where $\mathbf{g}$ maps frontier points to the view proposals discussed in Section 3.2.3.

Views are selected with the aim of obtaining a scene observation using the shortest overall travel distance by considering the *incremental* and *origin* distances for a set of view proposals. The incremental distance of a view proposal is given by the difference between the current view position and the position of the proposed view. The origin distance of a view proposal is given by the difference between the position of the initial view of the scene and the position of the proposed view. Reducing the incremental distance limits how far the sensor travels between views.

Accounting for the origin distance prioritises the selection of next best views that are close to the position of the initial view of the scene. This consideration of the origin distance is not mathematically founded but an empirical evaluation demonstrated that it successfully reduces the overall travel distance by penalising the selection of view trajectories that move away from the initial view in a singular direction. These trajectories often fail to fully observe the incomplete scene regions that surround the surfaces observed from the initial view and therefore the sensor has to return for measurements to be obtained, increasing the travel distance. The overall distance is reduced by selecting a next best view, $\mathbf{v}_{i+1}$, from a subset of view proposals, W' , within a given radius, η , of the current view, $\mathbf{v}_i = \{\mathbf{x}_i, \phi_i\}$,

$$W' = \{\mathbf{v} = \{\mathbf{x}, \phi\} \in W \mid \|\mathbf{x} - \mathbf{x}_i\| < \eta\}, \quad (3.17)$$

to minimise the origin distance from the position of the first view, $\mathbf{v}_0 = \{\mathbf{x}_0, \phi_0\}$,

$$\mathbf{v}_{i+1} = \arg \min_{\mathbf{v}=\{\mathbf{x}, \phi\} \in W'} (\|\mathbf{x} - \mathbf{x}_0\|). \quad (3.18)$$

If no view proposals exist within the specified radius of the current view (i.e., $W' = \emptyset$) then the next best view is selected to minimise the incremental distance,

$$\mathbf{v}_{i+1} = \arg \min_{\mathbf{v}=\{\mathbf{x}, \phi\} \in W} (\|\mathbf{x} - \mathbf{x}_i\|). \quad (3.19)$$

The presented metric for view selection aims to select next best views that will reduce the overall travel distance required to observe a scene. The goal for each view is to obtain a successful observation of a target frontier point (i.e., reclassify it as a core point). This is not always possible due to the presence of surface discontinuities and occlusions. In these cases the view is adjusted to try and obtain a successful observation of the scene surface beyond the discontinuity or around an occlusion.

3.2.5 View Adjustment

Most scenes contain discontinuous surface geometry and occluding surfaces. These characteristics often constrain the improvement in a scene observation that is achievable from a given view. Discontinuous surface geometry, such as the presence

of edges or corners, can prevent the coverage of a scene from being extended if a view is oriented to only observe the surface on one side of the discontinuity. An occlusion describes the scenario in which the observation of a target surface is obscured from a given view by the presence of another scene surface that is closer to the view.

Surfaces that are occluded or contain discontinuities that are not visible from a given view can be successfully observed by identifying an alternative view that is unoccluded and from which both sides of a discontinuity are visible. Strategies for identifying a view that is likely to obtain a successful observation can be *reactive* or *proactive*. Reactive strategies apply incremental adjustments to unsuccessful views based on the measurements obtained until a successful observation is achieved. Proactive strategies consider knowledge of the scene geometry obtained from previous views to try and propose a successful view. The value of proactively handling pointwise occlusions in an unstructured representation is investigated in Chapter 4.

In this chapter surface discontinuities and occlusions are addressed reactively. Views that do not obtain a successful observation of their target frontier points are incrementally adjusted until a successful observation is obtained or a termination criterion is satisfied. The adjustment of a view is computed as a sequence of transforms based on the distance and relative orientation of the target frontier point from the pointwise mean for point measurements obtained from the current view (Fig. 3.4). Incremental adjustments are applied to reduce the distance between the frontier and pointwise mean until a successful observation is obtained or the distance stops reducing. The adjustment then terminates and an alternative view is proposed.

The transformation of a view is performed in a coordinate frame defined by the orthogonal set of vectors computed from the local surface geometry estimate for the target frontier point, $\mathbf{R}_d = [\mathbf{e}_n \ \mathbf{e}_f \ \mathbf{e}_b]$. The magnitude of translation and rotation for each axis is determined by the distance, $\mathbf{s} := [s_0, s_1, s_2]^T$, between the pointwise mean for the observed points, $\boldsymbol{\omega}$, and the frontier point, $\mathbf{f}$, along the axis,

$$\mathbf{s} = \mathbf{R}_d^T (\mathbf{f} - \boldsymbol{\omega}). \quad (3.20)$$

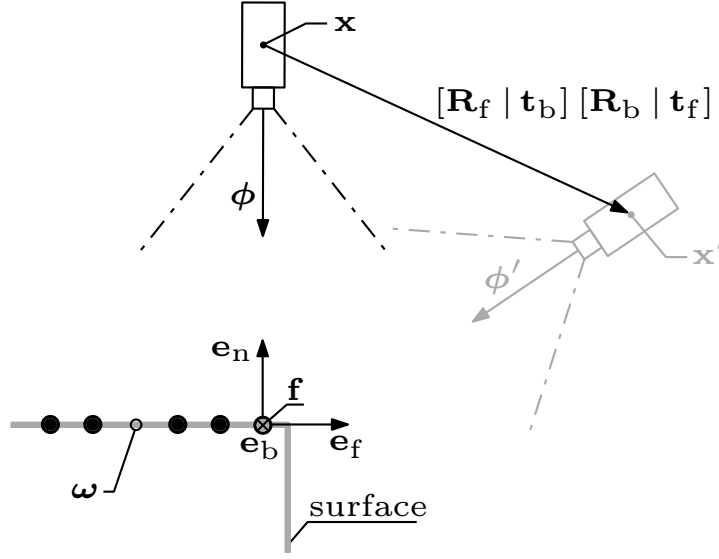


Figure 3.4: An illustration of a scenario where it is necessary to adjust the view of a target frontier point in order to obtain a successful observation. The frontier point, $\mathbf{f}$, lies on one side of a surface edge (grey line) and the current view, $\mathbf{v} = \{\mathbf{x}, \phi\}$, is orthogonal to the surface plane on that side. A successful observation of the frontier is not obtained from this view as the surface on the opposing side of the edge is not visible. An adjusted view, $\mathbf{v}' = \{\mathbf{x}', \phi'\}$, capable of viewing the surface beyond this discontinuity is obtained by applying transformations in a coordinate frame defined by the vectors $\mathbf{e}_n$, $\mathbf{e}_f$ and $\mathbf{e}_b$. The view is adjusted by rotations, $\mathbf{R}_b$ and $\mathbf{R}_f$, and translations, $\mathbf{t}_f$ and $\mathbf{t}_b$, computed using the frontier, $\mathbf{e}_f$, and boundary, $\mathbf{e}_b$, vectors. The adjustment magnitude is determined by a scaling factor, d_t , and the distance of the frontier point from the pointwise mean, ω , for observed points (black dots).

The translation vectors, $\mathbf{t}_f$ and $\mathbf{t}_b$, along each axis have a length given by a product of the corresponding distance, s_i , with a scalar, d_t . The rotation matrices, $\mathbf{R}_b$ and $\mathbf{R}_f$, are computed using Rodrigues' rotation formula (Rodrigues 1840).

A translation along the frontier vector is applied to move the centre of the viewing frustum in the direction of the partially observed region of the scene,

$$\mathbf{t}_f = s_1(d_t + 1)\mathbf{e}_f. \quad (3.21)$$

A rotation around the boundary vector is used to improve the visibility of a surface on the opposing side of a discontinuity or behind an occluding surface,

$$\mathbf{R}_b = \mathbf{I} + \sin \theta_b \mathbf{e}_b^\times + (1 - \cos \theta_b)(\mathbf{e}_b^\times)^2, \quad (3.22)$$

where

$$\theta_b = \tan^{-1} \left(\frac{ds_1 d_t}{d^2 + s_1^2 (d_t + 1)} \right) \text{ and } \mathbf{u}^\times = \begin{bmatrix} 0 & -u_2 & u_1 \\ u_2 & 0 & -u_0 \\ -u_1 & u_0 & 0 \end{bmatrix}. \quad (3.23)$$

A translation along the boundary vector moves the centre of the viewing frustum in the direction with fewer observed points (i.e., away from the pointwise mean),

$$\mathbf{t}_b = s_2(d_t + 1)\mathbf{e}_b. \quad (3.24)$$

This allows for the observation to be extended beyond multiple discontinuities in cases where the frontier lies near the intersection of multiple surfaces (e.g., a corner).

A rotation around the frontier vector is used to move the centre of the viewing frustum towards the point of intersection between multiple surfaces, if one exists,

$$\mathbf{R}_f = \mathbf{I} + \sin \theta_f \mathbf{e}_f^\times + (1 - \cos \theta_f)(\mathbf{e}_f^\times)^2, \quad (3.25)$$

where

$$\theta_f = \tan^{-1} \left(\frac{ds_2 d_t}{d^2 + s_2^2 (d_t + 1)} \right). \quad (3.26)$$

The distance factor, d_t , determines the magnitude of translation and rotation used for the view adjustment. It is scaled exponentially with the number of adjustments, n , for a given frontier, $d_t = 2^n$. This stops the magnitude from converging to zero as the pointwise mean moves closer to the frontier point.

The adjusted view, $\mathbf{v}' = \{\mathbf{x}', \phi'\}$, is computed from the current view position, $\mathbf{x}$,

$$\begin{aligned} \phi' &= \mathbf{f} - \mathbf{R}_f(\mathbf{t}_b + \mathbf{R}_b(\mathbf{t}_f + \mathbf{x})), \\ \mathbf{x}' &= \mathbf{f} - d \frac{\phi'}{\|\phi'\|}. \end{aligned} \quad (3.27)$$

This view is chosen to be the next best view instead of selecting another view proposal and new measurements are obtained. This process is repeated iteratively until the frontier point is successfully observed (i.e., the opposing sides of a surface discontinuity are observed or an occlusion is avoided) or the Euclidean distance between the frontier point and the pointwise mean for observed points stops reducing. If this termination criterion is reached then the view proposal is switched to the view from which the frontier point was first observed, $\mathbf{v}_o$, and the view adjustment

is restarted with the distance factor reinitialised to $d_t = 1$. If this adjustment also reaches the termination criterion then the frontier point is reclassified as an outlier.

The presented strategy for incrementally adjusting unsuccessful views of frontier points allows SEE to extend the observation of a scene beyond surface discontinuities and reactively avoid occluding surfaces. This enables SEE to obtain highly complete observations of scenes with discontinuous surface geometry and self-occlusions as the observation of a frontier point can be attempted from multiple views, including the view from which the frontier was first observed, until a successful observation is obtained. The termination criterion is defined to end an attempted observation when a frontier point is determined to be unobservable. This typically occurs when the measurement is a product of sensor noise rather than obtained from a true surface.

3.2.6 Completion

The observation of a scene is considered complete when there are no more frontier points remaining (i.e., all points are classified as either core points or outliers). This means that the desired measurement density has been achieved for all scene surfaces with core point measurements. Remaining outlier points are typically the product of sensor noise. The extent of an observation can be constrained by discarding point measurements outside of a user-specified bounding volume encompassing the scene.

3.3 Evaluation

SEE is compared with volumetric NBV planning approaches, (Area Factor (AF), Vasquez-Gomez et al. 2014; Average Entropy (AE), Kriegel et al. 2015; and Rear Side Voxel (RSV), Rear Side Entropy (RSE), Unobserved Voxel (UV), Proximity Count (PC), Occlusion Aware (OA), Delmerico et al. 2018) on four one-metre standard models (Newell Teapot, Newell 1975; Stanford Bunny, Turk and Levoy 1994; Stanford Dragon, Curless and Levoy 1996; and Stanford Armadillo, Krishnamurthy and Levoy 1996) and on a 40 metre model of the Radcliffe Camera (Boronczyk 2016). The implementations of the volumetric approaches are provided by Delmerico et al. (2018). Every algorithm was run to completion for 100 experiments on each model.

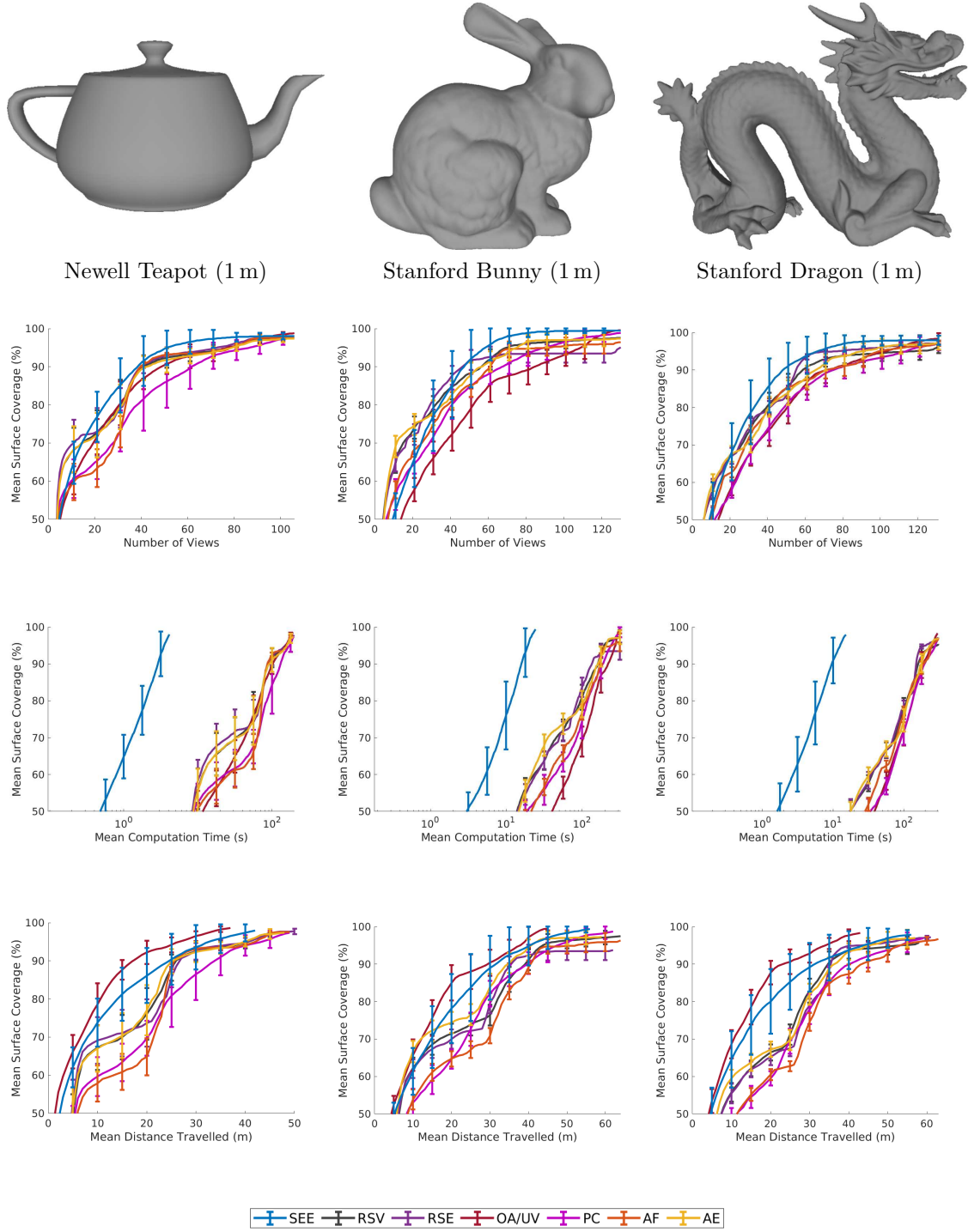


Figure 3.5: An experimental comparison of SEE with the evaluated volumetric approaches. The graphs show the mean surface coverage obtained by SEE and the volumetric approaches from 100 experiments relative to, from top to bottom, the number of views, the mean computation time and the mean travel distance.

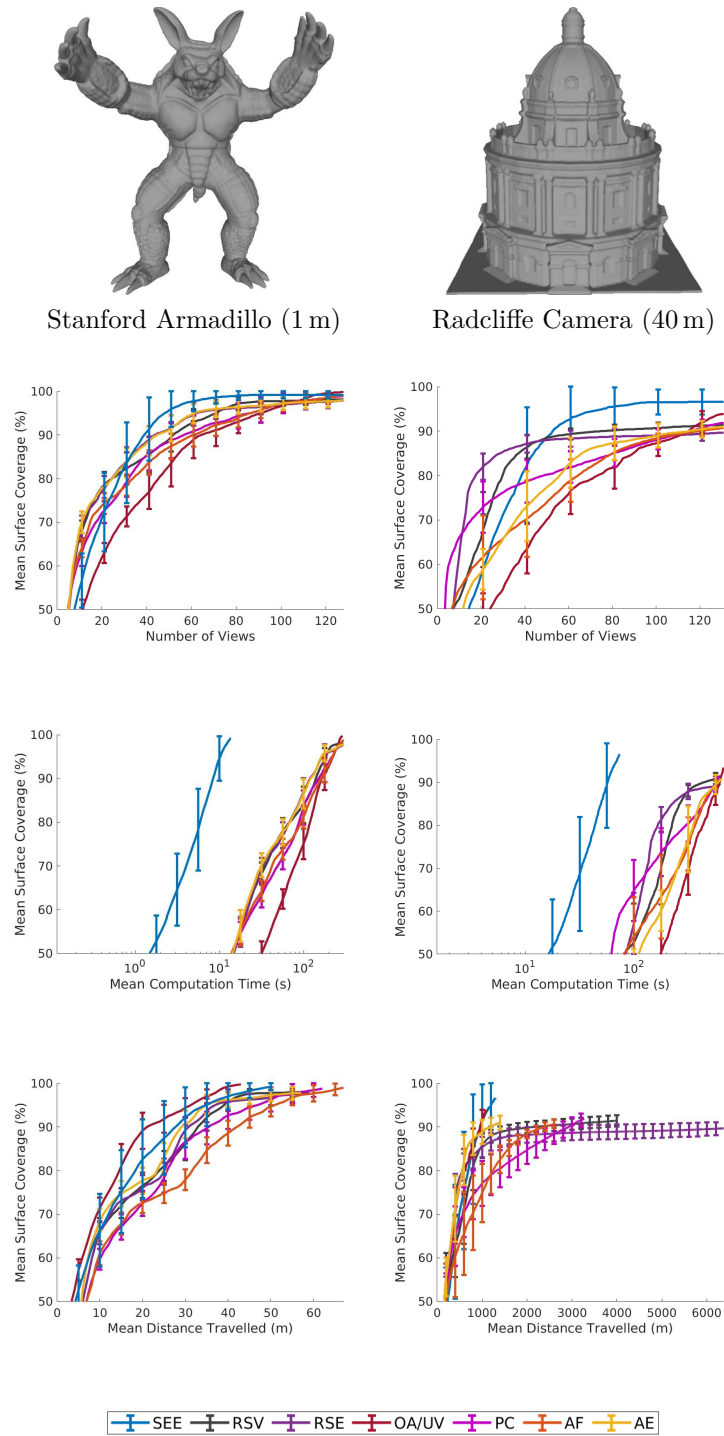


Figure 3.6: An experimental comparison of SEE with the evaluated volumetric approaches. The graphs show the mean surface coverage obtained by SEE and the volumetric approaches from 100 experiments relative to, from top to bottom, the number of views, the mean computation time and the mean travel distance.

	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	60.0	105	105	105	105	105	105	105
Surface Coverage (%)	98.1	97.6	97.8	98.8	97.6	98.8	97.6	97.4
Computation Time (s)	4.13	196	200	194	198	194	196	196
Distance Travelled (m)	42.8	49.6	50.7	37.8	50.0	37.8	49.9	48.5

(a) Newell Teapot (Newell 1975)

	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	75.3	129	129	129	129	129	129	129
Surface Coverage (%)	99.5	97.6	95.1	99.5	98.9	99.5	96.4	97.4
Computation Time (s)	24.7	325	324	313	320	313	326	324
Distance Travelled (m)	56.3	64.8	62.7	45.1	62.9	45.1	64.3	59.8

(b) Stanford Bunny (Turk and Levoy 1994)

	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	78.0	130	130	130	130	130	130	130
Surface Coverage (%)	98.0	96.1	97.2	98.4	97.2	98.4	97.1	97.3
Computation Time (s)	15.4	311	311	300	306	300	306	311
Distance Travelled (m)	56.8	58.2	61.1	43.1	59.2	43.1	63.8	58.0

(c) Stanford Dragon (Curless and Levoy 1996)

	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	63.4	127	127	127	127	127	127	127
Surface Coverage (%)	99.2	98.2	98.0	99.8	98.9	99.8	99.1	98.0
Computation Time (s)	13.7	298	301	291	298	291	298	301
Distance Travelled (m)	50.3	59.2	59.2	43.5	62.7	43.5	67.3	57.5

(d) Stanford Armadillo (Krishnamurthy and Levoy 1996)

	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	64.5	130	130	130	130	130	130	130
Surface Coverage (%)	96.6	91.4	89.7	93.9	91.9	93.9	90.7	91.0
Computation Time (s)	74.4	648	630	674	622	675	628	650
Distance Travelled (m)	1302	4008	6375	1098	3272	1098	2615	1409

(e) Radcliffe Camera (Boroczyk 2016)

Table 3.1: The mean number of views captured, the mean surface coverage obtained, the mean computation time used and the mean travel distance required to observe four one-metre standard models (Newell Teapot, Stanford Bunny, Stanford Dragon, and Stanford Armadillo) and a 40 metre model of the Radcliffe Camera, calculated from 100 experiments with SEE and the volumetric approaches.

	Intel Realsense D435	Simulated LiDAR
θ_x ($^\circ$)	69.4	60.0
θ_y ($^\circ$)	42.5	40.0
w_x (px)	848	2400
w_y (px)	480	1750

Table 3.2: The field-of-view in degrees, θ_x and θ_y , and resolution in pixels, w_x and w_y , of the simulated depth sensors used to obtain observations of the scene models.

3.3.1 Simulated Sensors

Point measurements from a simulated depth sensor are obtained by raycasting into the triangulated surface mesh of a scene model and adding Gaussian noise ($\mu = 0$ m, $\sigma = 0.01$ m) to the ray intersections. Simulated depth sensors are defined by a field-of-view in degrees, θ_x and θ_y , and a resolution in pixels, w_x and w_y . The simulation environment contains no ground plane and the sensor can move unconstrained in three dimensions with six degrees of freedom. The standard models are observed using a simulated Intel Realsense D435 and the Radcliffe Camera is observed with a simulated LiDAR sensor (Table 3.2).

3.3.2 View Constraints

View proposals for the volumetric approaches are sampled from a surface encompassing the scene, in this case a sphere, as presented by Vasquez-Gomez et al. (2014) and Delmerico et al. (2018). Kriegel et al. (2015) does not sample views from an encompassing view surface but we use the implementation provided by Delmerico et al. (2018) which does. The radius of the view sphere is set to the sum of the view distance, d , and a surface offset that accounts for the model size. It is equal to the mean distance of points in the model from their centroid.

In the experiments with SEE next best views are selected until its completion criterion is satisfied. The view limiting termination criterion used by volumetric approaches is then set to the maximum number of views obtained by SEE in any one experiment. The number of views sampled from the sphere encompassing the scene is defined as 2.4 times the view limit, as presented by Delmerico et al. (2018).

	Standard Models	Radcliffe Camera
ρ (points per m^3)	146000	213
r (m)	0.017	0.15
η (m)	$3r$	$3r$
d (m)	1.98	41.3
γ	$0.1w_xw_y$	$0.02w_xw_y$
ϵ	$\sqrt{\rho^{-1}}$	$\sqrt{\rho^{-1}}$

Table 3.3: The parameters used by SEE and the volumetric approaches to observe the one-metre standard models and the 40 metre model of the Radcliffe Camera.

3.3.3 Algorithm Parameters

The parameters used by SEE and the evaluated volumetric approaches to obtain observations of the small-scale standard models and the large-scale Radcliffe Camera model are shown in Table 3.3. The target measurement density, ρ , used by SEE is set to be sufficiently large that frontiers in the pointcloud observation can be reliably identified when using the chosen resolution radius, r . For the standard model experiments the resolution radius is selected to be large enough that the simulated sensor noise can be handled robustly while obtaining high-density observations. For the Radcliffe Camera experiments a larger resolution radius is used to account for the increased scene scale and the longer range at which measurements are obtained. The volumetric approaches use the same resolutions to define the size of their voxels.

The NBV search radius used by SEE in (3.17), η , was determined experimentally. The view distance, d , is computed from the target density and sensor parameters using (6.1). The raycasting resolution, γ , used by the volumetric approaches to evaluate voxel visibility is chosen to be small enough to attain a reasonable computation time without significantly reducing the observation performance.

In all experiments, a minimum separation, ϵ , between point measurements is enforced to maintain an upper bound on memory consumption and computational cost. It is computed from the target density as the separation that would occur if a number of points equal to the target density were uniformly distributed on the surface of a unit square. New measurements are only added to the pointcloud observation if their ϵ -radius neighbourhood contains no existing measurements.

3.3.4 Performance Metrics

The algorithms are evaluated by calculating the relative surface coverage, computational time and sensor travel distance. These values are averaged across the 100 experiments performed on each model using each of the approaches.

The surface coverage of an approach is measured as the ratio of observed model points, M_o , to total model points, M_t ,

$$\tau := \frac{M_o}{M_t}. \quad (3.28)$$

A point is considered observed, $M_o \subseteq M_t$, if there is a measurement within r_d of the point. This registration distance is chosen as $r_d = 0.005$ m for the standard models, as in Delmerico et al. (2018), and $r_d = 0.05$ m for the Radcliffe Camera model.

The time taken to compute next best views is measured and added to a cumulative total. The time required to travel between views is not considered.

The distance travelled by the sensor is measured by summing the Euclidean distance between the positions of subsequent views. This metric is used to quantify the observation cost in lieu of measuring the operation time or energy consumption required to move the sensor between views as these considerations are specific to a particular platform and not applicable to a simulated free-flying sensor.

3.4 Discussion

The improvement in observation performance achievable by using an unstructured representation that proposes and selects views based on the density of observed points is demonstrated by the experimental results (Fig. 3.5; Fig. 3.6; Table 3.1).

SEE is shown to obtain observations for all of the models with an equivalent or greater surface coverage than all of the evaluated volumetric approaches while using an order of magnitude lower computational time and fewer views in the mean case.

The computation time used by SEE is significantly lower as all updates are performed pointwise and locally while the volumetric approaches utilise raycasting to quantify the value of every sampled view before selecting a next best view.

Raycasting is computationally expensive as the number of evaluated voxels increases cubically with the scene scale, when using a fixed voxel size, and linearly with the raycasting resolution (i.e., the number of rays). The cost of this complexity when obtaining observations of large-scale scenes using a high resolution is demonstrated by an increase in the computational time when observing the Radcliffe Camera.

SEE is able to obtain highly complete observations using fewer views than the volumetric approaches by directly considering scene information when proposing views and evaluating observation completeness. Using knowledge of the scene structure to propose views means that the visibility of scene surfaces is not restricted by the number and distribution of a fixed set of views sampled around the scene. Quantifying the completion of an observation means that it is only necessary to capture a sufficient number of views to attain a given minimum measurement density.

The experimental results show that the volumetric approaches achieve high surface coverage for the standard models by sampling a high density of views. Similarly high coverage is not obtained when the same number of views are sampled to observe the large-scale Radcliffe Camera. The sparser distribution of views means that some surfaces are not visible from any sampled view and it is not possible to achieve visibility by adjusting views. SEE is capable of attaining high coverage for the Radcliffe Camera by proposing views to improve surface coverage and adjusting views when measurements around a target frontier are not successfully obtained.

The sensor travel distances required to obtain scene observations using SEE are lower than many of the volumetric approaches except for OA and UV. These approaches are able to observe scenes by travelling shorter distances than SEE while capturing more views as they consider occlusions. Both approaches weight the information gain associated with observing voxels from a given view by the probability that they are visible from the view. The probability that a target voxel is visible is computed from the occupancy likelihood of other voxels between the target voxel and the view. The NBV metrics used by OA and UV prioritise the selection of views with good visibility that are close to current sensor position. This reduces the overall travel distance as the sensor only travels farther distances when the visibility

of nearby views is occluded. The demonstrated value of considering occlusions before capturing views motivates the work on proactive occlusion handling with an unstructured scene representation that is presented in the proceeding chapter.

This chapter demonstrates the value of using an unstructured scene representation for NBV planning. The use of high-fidelity pointwise scene information makes it possible to propose views that can improve an observation by considering the scene structure. These views can be reactively adjusted in response to new measurements until a successful improvement in the scene observation is obtained. Next best views are selected and captured until a given measurement density is attained for all scene surfaces. Experimental results demonstrate that the resulting SEE approach is able to obtain highly complete scene observations using fewer views and a lower computational time than the evaluated state-of-art volumetric approaches.

This work provides the foundation for investigations into proactive occlusion handling (Ch. 4) and considering scene visibility (Ch. 5) with an unstructured scene representation. Solutions to these challenges have been presented for structured representations but it was necessary to formulate novel techniques suitable for an unstructured representation. The investigations aim to increase the observation performance of SEE by improving the visibility of scene surfaces in order to reduce the number of views and travel distance required to obtain scene observations.

In summary, the work presented in this chapter makes five key contributions:

1. A novel unstructured scene representation based on measurement density.
2. A method for proposing views directly from point-based scene information.
3. A novel technique for reactively handling occlusions that is capable of successfully observing target surfaces by using incrementally adjusted views.
4. A termination criterion that considers observation completeness by ensuring that a minimum measurement density is obtained for all observable surfaces.
5. Experimental results demonstrating that SEE is capable of obtaining scene observations with similar or better coverage than state-of-the-art volumetric approaches using fewer views and a significantly lower computation time.

4

Proactively Handling Occlusions

Contents

4.1	Existing Methods	71
4.2	Detecting Occlusions	74
4.2.1	Defining Visibility	74
4.2.2	Naive Search	75
4.2.3	Adaptive Search	76
4.3	Proposing Unoccluded Views	77
4.3.1	Representing Occlusions	78
4.3.2	Mean Strategy	79
4.3.3	Eigenvector Strategy	80
4.3.4	Geodesic Strategy	82
4.3.5	Optimisation Strategy	84
4.4	Evaluation	89
4.5	Discussion	89

This chapter presents an investigation of strategies for proactively handling occlusions with an unstructured scene representation. It is often necessary for NBV planning approaches to consider occlusions in order to obtain the most complete observation of a scene possible given sensing and termination constraints. A scene observation can be considered complete if the sensor measurements obtained cover every surface in the scene. This is only possible if every scene surface is visible from at least one of the views obtained. Occlusions can prevent the observation of a surface from certain views and potentially render a complete scene observation unobtainable.

A target surface is occluded from a given view if another surface exists within the view frustum of the sensor between the target surface and the view position. An occluding surface can be part of the scene (i.e., a self-occlusion) or another object within the sensor workspace. In order to successfully observe the target surface a view must be found that does not have an occluding surface within its frustum. Occlusions can be addressed *reactively* by capturing incrementally adjusted views until the target surface is observed or handled *proactively* by detecting known occlusions before a view is obtained and proposing an alternative view that is unoccluded.

SEE addresses occlusions reactively (Sec. 3.2.5). When a frontier point is not successfully observed from its associated view, the view is incrementally adjusted and new measurements are obtained until it is observed or determined to be unobservable. Unoccluded views can be found using this strategy but it is inefficient in terms of the distance travelled and number of views required to observe a frontier.

The work on proactive occlusion handling presented in this chapter aims to obtain unoccluded views more efficiently. Point-based occlusions are detected before views are obtained (Sec. 4.2) and encoded in a novel pointwise representation (Sec. 4.3.1). Several strategies for proposing unoccluded views using the information encoded in this representation are investigated (Sec. 4.3). The observation performance of these strategies is evaluated experimentally in comparison with SEE (Sec. 4.4).

The work presented in this chapter makes three key contributions:

1. A computationally efficient method for detecting point-based occlusions.
2. A novel representation for encoding pointwise occlusion information.
3. An investigation of several strategies for proposing unoccluded views.

4.1 Existing Methods

Existing methods for detecting occlusions almost exclusively use *raycasting*. This technique computes the intersection of a ray, defined by an origin and direction, with any two-dimensional (e.g., a plane) or three-dimensional (e.g., a cube) manifolds that collide with the ray in real space. Volumetric approaches use raycasting to detect collisions between a set of rays, with origins at the view position and

directions within the view frustum, and voxels in their grid representation. Surface approaches use raycasting to detect collisions between a set of rays and triangles in their mesh representation. The visibility of a represented manifold (e.g., a voxel or surface triangle) from a given view is considered fully occluded if every ray intersecting it collides with another manifold first or partially occluded if some of the intersecting rays have no previous collisions. A represented manifold is fully visible if none of its intersecting rays have previous collisions.

The unstructured density representation used in SEE only contains observed points (i.e., zero-dimensional manifolds) with which ray intersections can not be computed. In order to use raycasting with this representation it would be necessary to manifest the points as three-dimensional spheres by augmenting the point representation with a radius parameter. This would require a heuristic determination of the radius parameter that may not generalise between scenes. Too large a value would result in false positive detections and too small a radius could produce false negatives. It could also be computationally expensive as in the worst case a naive raycasting method may evaluate the intersection of m rays with n augmented points for a complexity of $O(mn)$. Raycasting is therefore not a suitable method for detecting point-based occlusions with an unstructured scene representation.

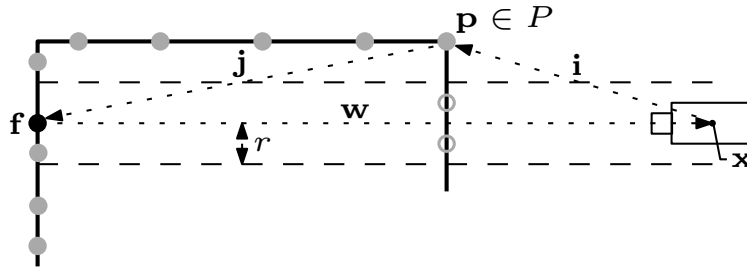
Katz et al. (2007) present Hidden Point Removal (HPR), an approach for determining the visibility of points from a given view. HPR can identify the set of points in a pointcloud that are visible from a specified view. Points are projected using the *spherical flipping* technique presented by Katz et al. (2005). The projection of each point is computed relative to the surface of a sphere centred on the view position and with a sufficiently large radius to encompass the pointcloud. The spherical flipping preserves the orientation of projected points relative to the view and determines their position as a ratio of the distance from the view to the original point and the sphere radius. The projection function is monotonically decreasing such that points closer to the view are projected farther from the sphere. A convex hull is computed from the set of projected points and the view position. Points whose projections are in the convex hull are determined to be visible from the view.

HPR does provide a solution for detecting point-based occlusions. It requires no heuristic augmentation of the pointwise representation and has a computational complexity of $O(n \log n)$ (Katz et al. 2007). This would provide greater efficiency than using a naive raycasting method if the number of observed points, n , is not logarithmically greater than the number of rays. If the objective was to identify the set of all points that are visible from a given view then HPR would be the ideal solution. However, in this scenario it is only necessary to consider a single point so assessing the visibility of other points introduces unnecessary complexity. HPR is therefore not considered to be an ideal solution for detecting point-based occlusions.

Most existing NBV planning approaches do not proactively handle occlusions by proposing unoccluded views based on existing scene information. Approaches that sample a fixed set of views from a surface encompassing the scene usually perform an exhaustive search to determine the visibility of every manifold in the scene representation from each view. This method is computationally expensive and will not obtain an unoccluded view if one does not exist in the fixed view set.

Approaches that consider visibility when sampling views from the scene volume using path planning techniques typically either sample views in free space until a given visibility criterion is satisfied (Song and Jo 2017) or sample them from a region with known visibility (Song and Jo 2018). Approaches that propose views based on scene observations often apply reactive strategies (Kriegel et al. 2015), similar to the one used by SEE, that incrementally adjust views until a successful observation is obtained or the adjustment exceeds a given threshold. As discussed previously this is an inefficient strategy as the incremental views obtained typically do not significantly improve the scene observation relative to the travel distance and time required.

Given that existing solutions for considering occlusions are not suitable for use with an unstructured scene representation it was necessary to investigate novel point-based strategies for detecting occlusions and proposing unoccluded views. To the best of my knowledge this is the first work to present solutions for proactively handling occlusions with an unstructured density representation. It is unique in its direct consideration of occlusions when proposing views using scene information.



4.2 Detecting Occlusions

4.2.1 Defining Visibility

$$D(\mathbf{f}, \mathbf{v}) := \left\{ \mathbf{p} \in P \mid \left\| \mathbf{j} - \mathbf{w} \frac{\mathbf{w} \cdot \mathbf{j}}{\|\mathbf{w}\|} \right\| \leq r \wedge \mathbf{w} \cdot \mathbf{i} < 0 \wedge \mathbf{w} \cdot \mathbf{j} < 0 \right\}, \quad (4.1)$$

where $\mathbf{a} \cdot \mathbf{b}$ denotes the dot product of two vectors, P is the set of all observed points, $\mathbf{w} = \mathbf{x} - \mathbf{f}$ is the sight line, $\mathbf{i} = \mathbf{p} - \mathbf{x}$ is a vector from the view position to a point and $\mathbf{j} = \mathbf{f} - \mathbf{p}$ is a vector from a point to the frontier. The constraints $\mathbf{w} \cdot \mathbf{i} < 0$ and $\mathbf{w} \cdot \mathbf{j} < 0$ ensure that only points between the view position and frontier are considered. An empty set of occluding points, $D \equiv \emptyset$, denotes that the frontier point is visible from the view and it is otherwise occluded.

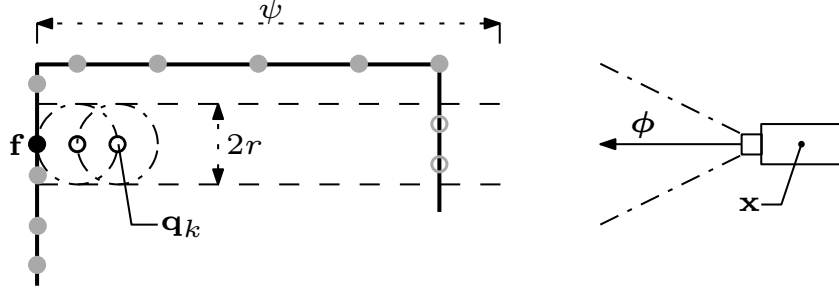


Figure 4.2: An illustration of how frontier visibility is determined using the naive search-based approximation. Occluding points (grey circles) are identified within the r -radii of search points (black circles), $\mathbf{q}_k \in Q$, sampled along the sight line from the frontier point (black dot), $\mathbf{f}$, to the view, $\mathbf{v} = \{\mathbf{x}, \phi\}$, up to a given distance, ψ .

4.2.2 Naive Search

As the number of point measurements increases, evaluating whether every observed point is within an r -radius of the sight line between a frontier point and its associated view becomes computationally expensive. This complexity is reduced by using a naive search-based approximation (Fig. 4.2) to identify occluding points within the r -radii of a set of points, Q , sampled at an r -interval along the sight line,

$$Q := \left\{ \mathbf{f} + kr \frac{\mathbf{w}}{\|\mathbf{w}\|} \mid k = 1, 2, \dots, \frac{\psi}{r} \right\}, \quad (4.2)$$

starting at an offset from the frontier and up to a given occlusion search distance, ψ .

The first point is sampled with an offset of $k = 1$ from the frontier point to ensure that points on the same surface and behind the frontier are not identified as occlusions. This assumes that the sight line is orthogonal to the local surface.

The last point is sampled before a specified occlusion search distance is exceeded. In an ideal scenario this is equal to the view distance, $\psi = d$, but it is provided as a user-defined parameter so the computational cost of handling occlusions, including proposing unoccluded views, can be managed for computationally expensive observations (e.g., when observing large-scale scenes with long range sensors).

Occluding points are identified within the r -radius of each sampled point, $\mathbf{q} \in Q$,

$$N(P, r, \mathbf{q}) := \{\mathbf{p} \in P \mid \|\mathbf{p} - \mathbf{q}\| \leq r\}, \quad (4.3)$$

such that set of occluding points between the frontier point and view is the union of the sets of neighbouring points within the r -radius of each sampled point,

$$D'(\mathbf{f}, \mathbf{v}) := \bigcup_{\mathbf{q} \in Q} N(P, r, \mathbf{q}). \quad (4.4)$$

An empty set of occluding points, $D' \equiv \emptyset$, denotes that the frontier point is visible from the view and is otherwise occluded, as specified in the visibility definition.

4.2.3 Adaptive Search

The naive search method relies on an assumption that the sight line between a frontier point and its view is orthogonal to the local surface when defining an offset for sampling the first search point. This is an invalid assumption in many cases, particularly when assessing the visibility of a frontier point from a different view than the one originally proposed to observe it (i.e., the view obtained using an estimate of the local surface geometry). A better solution is to compute a variable offset, ζ , for each frontier point from existing visibility information (Fig. 4.3). The adaptive occlusion search replaces the fixed starting point (4.2) with this variable offset,

$$Q' := \left\{ \mathbf{f} + kr \frac{\mathbf{w}}{\|\mathbf{w}\|} \mid k = \zeta, \zeta + 1, \dots, \frac{\psi}{r} \right\}. \quad (4.5)$$

A suitable offset is determined by finding the first sampled point, $\mathbf{q}_k \in Q_o$, along the sight line between a frontier point and its observing view (i.e., the view from which the point was initially observed), $\mathbf{v}_o$, with no occluding points, $N(P, r, \mathbf{q}_k) \equiv \emptyset$, and using that offset, $\zeta = k$. This leverages the knowledge that as the point was originally observed from this view it is guaranteed to have been visible, provided the sensor measurement was obtained from a real surface (i.e., it is not the product of sensor noise), and so the presence of any potential occlusions closer than the offset did not prevent visibility of the surface. The assumption is made that this distance within which neighbouring points are not occluding also applies to other views.

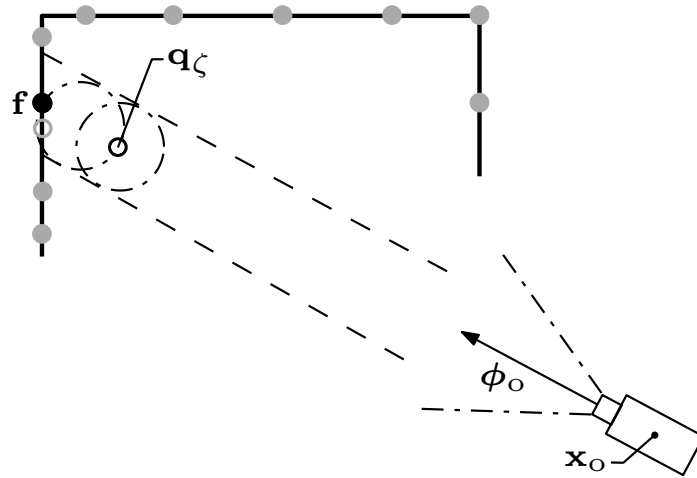


Figure 4.3: An illustration of a motivating scenario for using an adaptive offset, ζ , with the occlusion search and how a suitable offset is found. This shows that a frontier point (black dot) can be obtained from a surface despite the presence of a potentially occluding point (grey circle) within the first search radius. A suitable offset is determined by searching the sight line from the frontier to its observing view, $\mathbf{v}_o = \{\mathbf{x}_o, \phi_o\}$, and choosing the first sampled point for which no occlusions are found (black circle), $\mathbf{q}_\zeta$.

4.3 Proposing Unoccluded Views

When the visibility of a frontier point from its associated view is determined to be occluded it is necessary to find an unoccluded view from which a successful observation can be obtained. This can be achieved by identifying the directions from which a view of the frontier would be occluded and using known visibility information (e.g., the observing view) to estimate the direction of an unoccluded view. The observing view is not chosen even though it is unoccluded as capturing multiple measurements from the same view does not typically improve a scene observation.

This section presents a spherical projection for representing the directions from which occluding points would prevent the observation of a frontier. Different strategies are investigated that aim to use this representation and the observing view orientation, ϕ_o , to propose an unoccluded view orientation, ϕ' , with a corresponding view position at the view distance from the frontier point, $\mathbf{x}' = \mathbf{f} - d\phi'$.

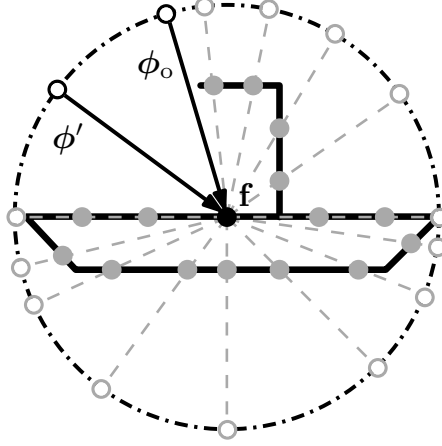


Figure 4.4: A cross-sectional illustration of the spherical projection used to represent sight lines along which the observation of a frontier point (black dot), $\mathbf{f}$, will be occluded. Points (grey dots) within the occlusion search distance of the frontier are projected (grey circles) onto a unit sphere centred on the frontier point based on their relative orientation to the sphere centre. Strategies for proposing unoccluded views aim to use the occlusion information encoded in this representation and the known visibility of the observing view orientation, ϕ_o , to propose an unoccluded view orientation, e.g., ϕ' , from which the frontier point can be successfully observed.

4.3.1 Representing Occlusions

The visibility of a frontier point from a view is considered to be occluded if observed points exist within a given distance of the sight line from the frontier point to the view (Sec. 4.2.1). The relative positions of observed points within a given distance of the frontier therefore denote sight lines along which a view at the specified distance or greater would be occluded. The orientation of these occlusions relative to the frontier point can be transformed into standardised representation by projecting points within the occlusion search distance, ψ , of the frontier, $\mathbf{f}$, onto a unit sphere with a centre equal to the frontier point (Fig. 4.4),

$$S = \left\{ \frac{\mathbf{p} - \mathbf{f}}{\|\mathbf{p} - \mathbf{f}\|} \mid \mathbf{p} \in N(P, \psi, \mathbf{f}) \right\}. \quad (4.6)$$

This representation is inspired by the spherical flipping technique used in HPR.

The successful identification of occluded sight lines using this spherical projection is dependent on the measurement accuracy of occluding points. Noisy sensor measurements obtained from surfaces close to the frontier point often deviate from the underlying surface and their projections can produce false occlusions. This

effect is mitigated by offsetting the projection centre, $\mathbf{c}$, of the unit sphere from the frontier point along the sight line to its observing view, $\mathbf{w}_o$, which is known to be unoccluded, using the offset computed by the adaptive occlusion search (Sec. 4.2.3),

$$\mathbf{c} = \mathbf{f} + \zeta r \frac{\mathbf{w}_o}{\|\mathbf{w}_o\|} . \quad (4.7)$$

The projection of occluding points is then computed relative to this offset centre,

$$S' = \left\{ \frac{\mathbf{p} - \mathbf{c}}{\|\mathbf{p} - \mathbf{c}\|} \mid \mathbf{p} \in N(P, \psi, \mathbf{f}) \right\} . \quad (4.8)$$

The following subsections present strategies that aim to use the occlusion information encoded in this representation and existing visibility knowledge for the frontier (i.e., its observing view) to estimate the orientation of an unoccluded view.

4.3.2 Mean Strategy

A simple strategy for estimating the orientation of an unoccluded view is to consider a sight line from the pointwise mean of the projected points that intersects the centre of the spherical projection. The position of this pointwise mean within the sphere relative to the centre indicates the direction of a region on the sphere containing the greatest number of projected points. A sight line from this position that intersects the sphere centre will therefore point away from the occupied region of the sphere and for simple occlusion configurations will intersect the sphere in a region of free space. A view along this sight line has orientation equal to the pointwise mean,

$$\phi' = \bar{\mathbf{s}} = \frac{1}{|S'|} \sum_{\mathbf{s} \in S'} \mathbf{s} . \quad (4.9)$$

This strategy can successfully obtain an unoccluded view for simple occlusion configurations but relies on an assumption that the projected points have an approximately uniform distribution over a contiguous region of the sphere. When the density of points in one region of the sphere is significantly greater than elsewhere the pointwise mean will be biased towards this region and the sight line from its position through the sphere centre may intersect an occupied region with fewer projected points (Fig. 4.5). If there exist discontinuous sets of projected points on opposing regions of the sphere then the pointwise mean will lie on the axis between the two regions and the sight line will intersect the region with fewer projected points.

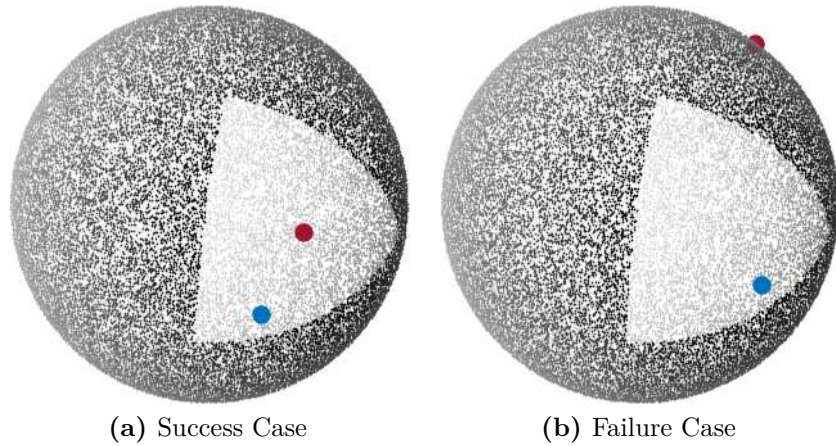


Figure 4.5: These visualisations show scenarios in which the mean strategy is (a) successful and (b) unsuccessful at identifying a sight line free from occlusions by using the spherical representation of occluding points (Sec. 4.3.1). The tiny black dots on the sphere surfaces represent the projections of virtual occluding points (not shown) that would obscure the visibility of a frontier point placed at the sphere center (i.e., they are equivalent to the grey circles shown in Fig. 4.4). The triangular regions containing no projected points represent space that is free from occlusions. The large blue dots denote the sight line of an observing view from which the frontier point at the sphere center would have been initially observed. This sight line is known to be unoccluded and is used by some strategies to inform the direction of a proposed view (e.g., by the eigenvector strategy) or as an initial solution for view proposals (e.g., by the geodesic and optimisation strategies). The large red dots represent the sight lines proposed by the mean strategy. In (a) the distribution of projected points is approximately uniform and the mean strategy successfully proposes an unoccluded view. In (b) this uniform distribution is augmented by adding additional points to the righthand side of the sphere. This biases the mean of the projected points and the mean strategy fails to propose an unoccluded view.

4.3.3 Eigenvector Strategy

The failure cases of the mean strategy can be partially mitigated by accounting for the variation in a set of projected points using Principal Component Analysis (PCA) (Pearson 1901; Hotelling 1933). PCA obtains a set of orthogonal basis vectors that represent the principal axes which account for the greatest amount of variance in a set of data. The axis associated with the least variation in the distribution of projected points is most likely to intersect a region of free space on the spherical projection. No direction is explicitly associated with this axis and therefore the computed view orientation is chosen to lie in the same hemisphere as the observing view.

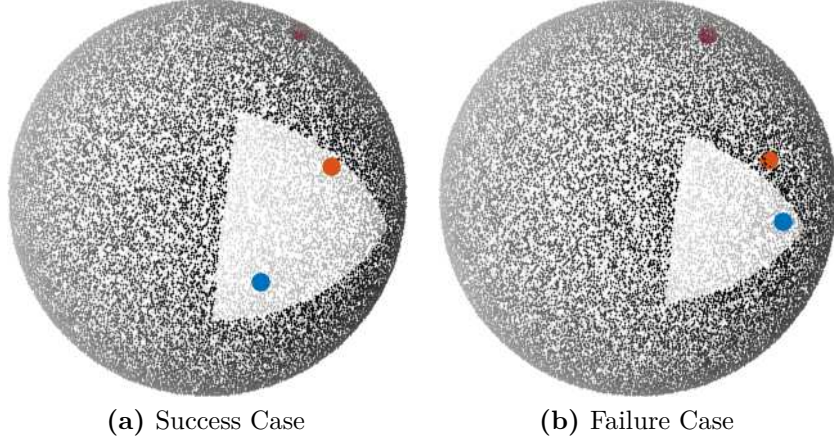


Figure 4.6: An experimental comparison of the sight lines proposed using the eigenvector strategy (orange dots) for different sized regions of free space on the spherical projection. Projected points are uniformly sampled over the occupied region of each sphere and then augmented with additional points sampled on the righthand side, as in Figure 4.5b. In (a) the eigenvector strategy successfully obtains an unoccluded view in a large region of free space when the mean strategy (red dots) fails as it is more robust to the nonuniform point distribution. In (b) the strategy fails for a smaller region of free space as the magnitude of bias on the axis of least variance for the projected points increases. The proposed view is set to point in the same direction as the observing view (blue dots) such that $\phi' \cdot \phi_o \geq 0$.

The set of orthogonal basis vectors is obtained by performing an eigenvalue decomposition on a matrix representation of the projected points, $\mathbf{s}_i \in S'$,

$$\mathbf{D} := [\mathbf{s}_1 - \bar{\mathbf{s}}, \dots, \mathbf{s}_n - \bar{\mathbf{s}}] \in \mathbb{R}^{3 \times |S'|}. \quad (4.10)$$

The decomposition of the covariance matrix, $\mathbf{A} := \mathbf{D}\mathbf{D}^T$, produces eigenvalues, $\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ and eigenvectors, $\Upsilon = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, which satisfy the eigenequation,

$$\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i, \quad i = \{1, 2, 3\}. \quad (4.11)$$

The view orientation is assigned as the eigenvector corresponding to the minimum eigenvalue (i.e., the axis associated with the least variation in the projected points),

$$\phi' = \{\mathbf{v}_i \mid \lambda_i = \min \{\Lambda\}\}, \quad (4.12)$$

and points in the same direction as the observing view (i.e., $\phi' \cdot \phi_o \geq 0$).

This strategy is more likely to obtain an unoccluded view for occlusion configurations with a nonuniform distribution of projected points than the mean strategy

provided there exists a sufficiently large region of free space on the spherical projection (Fig. 4.6). Accounting for the variance of projected points rather than just the pointwise mean allows an unoccluded view to be obtained for occlusion configurations with discontinuous sets of opposing points. The robustness of this strategy to nonuniform point distributions is limited as the covariance matrix used for the eigenvalue decomposition depends on the mean of the projected points.

4.3.4 Geodesic Strategy

An unoccluded view orientation can be obtained with greater success if the distribution of projected points is evaluated on the Riemannian manifold of the sphere instead of in Cartesian space. On this manifold the directions of greatest variation in a set of projected points can be represented with an orthogonal set of *principal circles* on the sphere. If a circle is *geodesic* (i.e., it denotes the shortest path between two points) then it defines a *great circle* and otherwise represents a *small circle*.

The set of principal geodesics that best represent the variance of points on a sphere can be obtained using the Principal Geodesic Analysis (PGA) approach presented by Fletcher et al. (2004), an adaption of the PCA technique for Riemannian manifolds. PGA is generalised by Jung et al. (2012) to Principal Nested Spheres (PNS), which obtains a set of principal circles that are not required to be geodesic. The geodesic strategy applies a subset of PNS, Principal Nested Great Spheres (PNGS), which uses the same method as PNS but only considers great circles.

The geodesic path along which there is the greatest variation in projected points is found by computing the closest possible great circle to the set of points. This path is the first principal circle computed using PNGS. The distance of projected points from a potential solution (i.e., a great circle) is computed and minimised in a tangent space of the sphere. The point of tangency defining this space is given by the intersection of the great circle axis with the sphere. As this axis is orthogonal to the geodesic path representing the greatest variance in projected points it denotes a vector along which there is minimal variation in the set of projected points. This therefore provides a good estimate for the orientation of an unoccluded view.

The axis of a great circle is represented by a vector pointing from the sphere centre along the axis. The point at which this vector intersects the sphere, $\mathbf{y}$, is used as a point of tangency to define a tangent space for the sphere. Projected points, $\mathbf{s} = [s_1, s_2, s_3] \in S'$, are mapped into this tangent space by a logarithmic function,

$$\text{Log}_{\mathbf{y}}(\mathbf{s}) = \left[s_1 \cdot \frac{\arccos(s_3)}{\sin(\arccos(s_3))}, s_2 \cdot \frac{\arccos(s_3)}{\sin(\arccos(s_3))} \right]. \quad (4.13)$$

The initial point of tangency is given by the intersection of the sight line for the observing view with the sphere. This is equivalent to the reverse of the observing view orientation as it is a unit vector, $\mathbf{y} = -\phi_o$.

Points in a tangent space of the sphere, $\mathbf{t} = [t_1, t_2]$, can be mapped back onto the sphere using an exponential function,

$$\text{Exp}_{\mathbf{y}}(\mathbf{t}) = \left[t_1 \cdot \frac{\sin \|\mathbf{t}\|}{\|\mathbf{t}\|}, t_2 \cdot \frac{\sin \|\mathbf{t}\|}{\|\mathbf{t}\|}, \cos \|\mathbf{t}\| \right]. \quad (4.14)$$

A solution point in the tangent space, $\mathbf{t}^*$, that is the shortest distance from the set of projected points is found by solving a least-squares minimisation problem (e.g., using the Levenberg-Marquardt algorithm; Levenberg 1943; Marquardt 1963),

$$\mathbf{t}^* = \arg \min_{\mathbf{t} \in \mathbb{R}^2} \sum_{\mathbf{s} \in S'} \left(\|\text{Log}_{\mathbf{y}}(\mathbf{s}) - \mathbf{t}\| - \frac{\pi}{2} \right)^2, \quad (4.15)$$

where the computed principal circle is constrained to be geodesic by requiring its radius to be $\frac{\pi}{2}$ and the initial solution point in the tangent space is $\mathbf{t} = [0, 0]$.

The point of tangency is updated with the solution point on the sphere, $\mathbf{y} \leftarrow \text{Exp}_{\mathbf{y}}(\mathbf{t}^*)$, and another iteration of the least-squares minimisation is performed. This continues until the solution point has converged to within a specified tolerance.

The solution point on the sphere defines the axis of the first principal circle along which the distance of projected points from the corresponding great circle has been minimised. Its position indicates the mostly likely direction of an unoccluded sight line from the frontier point as the initial solution point on the sphere is given by the known unoccluded sight line of its observing view. The proposed view orientation is therefore equal to the antipole of the solution point on the sphere, $\phi' = -\text{Exp}_{\mathbf{y}}(\mathbf{t}^*)$.

This strategy is capable of proposing unoccluded views with greater reliability and for more complex occlusion configurations than the mean and eigenvector

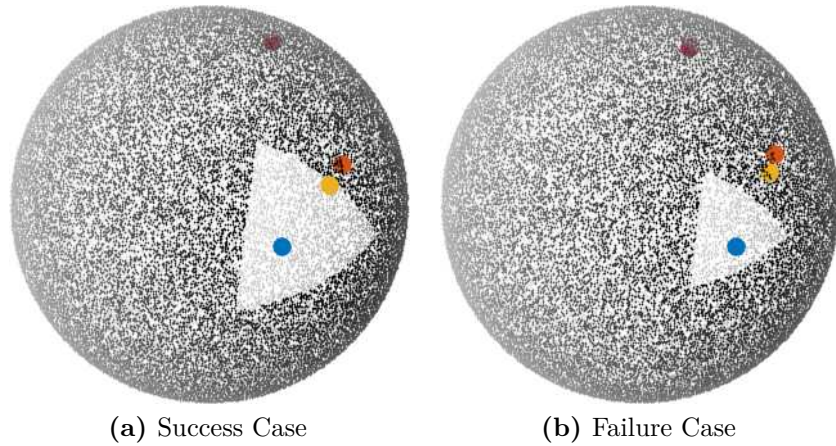


Figure 4.7: An experimental comparison of the sight lines proposed using the geodesic strategy (yellow dots) for different sized regions of free space on the spherical projection. Projected points are sampled on the sphere as in Figure 4.6. In (a), the geodesic strategy successfully obtains an unoccluded view in a region of free space when the eigenvector strategy (orange dots) fails as the biasing effect of the nonuniform distribution is reduced when considering variance on the sphere. In (b), this strategy fails when the region of free space is reduced as the biasing effect of the nonuniform distribution on the least-squares convergence increases. The initial solution point on the sphere is defined by the observing view (blue dots).

strategies. Accounting for the distribution of projected points on the Riemannian manifold of the sphere rather than in Cartesian space makes the estimate of an unoccluded view orientation more robust to a nonuniform distribution of projected points (Fig. 4.7). This approach is not capable of proposing an unoccluded view orientation for all occlusion configurations. For example, a nonuniform point distribution can produce an occluded view orientation by biasing the convergence of the least-squares minimisation to a solution point in an occupied region of the sphere.

4.3.5 Optimisation Strategy

Distribution-based strategies for proposing an unoccluded view (i.e., the mean, eigenvector and geodesic strategies) evaluate the occupied region of the spherical projection, by accounting for the distribution of projected points, to estimate a view orientation that will intersect a region of free space. An alternative approach to this problem is to directly identify the boundary between occupied and unoccupied regions of the sphere. An unoccluded view orientation can then be proposed to

intersect the centre of the unoccupied region. This view will have the greatest angular separation from nearby occlusions and therefore the best visibility of the target frontier point. The challenge of identifying this boundary between unoccupied and occupied regions of the sphere can be formulated as an optimisation problem.

The *maximin* facility location problem on a sphere, referred to in this thesis as the maximin optimisation problem, aims to find the location on a sphere that maximises the minimum distance to a set of points. There exists a complementary problem, the *minimax* facility location problem, which aims to find the location on a sphere that minimises the maximum distance to a set of points.

Drezner and Wesolowsky (1983)¹ present approaches for finding locally and globally optimal solutions to the minimax optimisation problem on a sphere and prove that the antipole of a minimax point is a solution to the complementary maximin optimisation. A locally optimal minimax solution is computed with a steepest descent algorithm. A global minimax solution is found by iteratively obtaining local solutions and checking their global optimality. This approach is not formulated to be solvable using standard nonlinear optimisation techniques and therefore the problem was investigated further to identify a nonlinear optimisation solution.

Patel and Chidambaram (2002) present a solution to the minimax problem formulated as a nonlinear optimisation with a linear objective function constrained by quadratic and linear constraints that can be solved using standard optimisation algorithms (e.g., Sequential Least-Squares Quadratic Programming (SLSQP); Kraft 1988; Kraft 1994). This approach obtains a minimax solution by finding the smallest spherical cap that encompasses every point on the sphere. Different methods are presented for handling hemispherical and spherical point distributions. The optimisation for hemispherical distributions is a convex problem for which a globally optimal solution is obtained. The optimisation for spherical distributions is nonconvex so a computed solution is not guaranteed to be globally optimal.

This thesis uses the approach presented by Patel and Chidambaram (2002) to find the view orientation which maximises the minimum distance to the set of projected

¹Drezner and Wesolowsky (1983) suggest an application for the maximin problem is finding the “location of a facility as far as possible from a given set of missile bases”.

points. A solution to the minimax problem is computed for which the corresponding maximin solution is the antipole, as shown by Drezner and Wesolowsky (1983).

The minimax solution is the centre of the smallest spherical cap that can contain all of the projected points. This cap is defined by a plane intersecting the sphere. The solution is found by optimising the orientation of the plane normal, $\mathbf{n}$, and its distance from the centre of the sphere, e . The plane normal points towards the smaller of the two spherical caps defined by the plane intersection. It is initialised using the orientation of the view from which the frontier point was first observed, ϕ_o , as this sight line is known to be unoccluded.

The specific optimisation method depends on the distribution of the projected points. If they are spread over the full sphere then the smallest containing cap will be larger than a hemisphere and is found by minimising the distance of the plane from the sphere centre,

$$\begin{aligned} (\mathbf{n}^*, e^*) &:= \arg \min_{\mathbf{n} \in \mathbb{R}^3, e \in [0,1]} e \\ \text{subject to} \quad & e \leq \mathbf{n}^T \mathbf{n}, \\ & e \geq \mathbf{n}^T \mathbf{s}_i, \quad i = 1, \dots, |S'|, \end{aligned} \tag{4.16}$$

where $\mathbf{s}_i \in S'$ is a projected point on the sphere. The initial distance is one and the initial normal is the inverse of the observing view orientation, $\mathbf{n} = -\phi_o$. The minimax solution, $\mathbf{s}'$, is the intersection of the inverse normal with the sphere,

$$\mathbf{s}' = -\frac{\mathbf{n}^*}{\sqrt{e^*}}, \tag{4.17}$$

as the normal points away from the containing cap.

If the projected points lie on a hemisphere then the full sphere optimisation converges to a plane bisecting the sphere (i.e., $e^* = 0$). This indicates the smallest containing cap must be smaller than a hemisphere. It can then be found by maximising the distance of the plane from the sphere centre,

$$\begin{aligned} (\mathbf{n}^*, e^*) &:= \arg \max_{\mathbf{n} \in \mathbb{R}^3, e \in [0,1]} e \\ \text{subject to} \quad & e \geq \mathbf{n}^T \mathbf{n}, \\ & e \leq \mathbf{n}^T \mathbf{s}_i, \quad i = 1, \dots, |S'|, \end{aligned} \tag{4.18}$$

where the initial distance is zero and the initial normal is the observing view orientation, $\mathbf{n} = \phi_o$. The minimax solution is the intersection of the plane normal with the sphere,

$$\mathbf{s}' = \frac{\mathbf{n}^*}{\sqrt{e^*}}, \quad (4.19)$$

as the normal points towards the containing cap.

The maximin solution is the antipole of the minimax solution. It represents the direction of an unoccluded sight line starting at the frontier point and pointing towards free space. This means the orientation of the view proposed to observe the frontier along this line is equal to the minimax solution, $\phi' = \mathbf{s}'$.

This strategy aims to identify a unoccupied region of the sphere by finding the smallest spherical cap which encompasses all of the projected points, such that the opposing cap will be a region of free space. A view is proposed to intersect the centre of the unoccupied cap as this will provide the greatest angular separation from occlusions. When the projected points lie on a hemisphere the optimisation problem is convex and a globally optimal solution can be guaranteed (i.e., the proposed view will be unoccluded and have the greatest possible separation from known occlusions).

If projected points are distributed over the full sphere then the problem is nonconvex and a globally optimal solution is not guaranteed. The solution may converge to a local minima on an occupied region of the sphere resulting from a nonuniform point distribution and produce an occluded view (Fig. 4.8). These failures are nullified by defining the initial solution as the sight line of the observing view, which is known to be unoccluded. Ensuring that the initial solution lies in free space guarantees that the optimisation will converge to a local minima within the same unoccupied region and produce an unoccluded view proposal (Fig. 4.9).

The guarantee that the sight line of the proposed view will be free from known occlusions is a significant improvement on distribution-based strategies which depend on evaluating the distribution of projected points to estimate a view orientation but provide no guarantee that it will be unoccluded. However, proposing an unoccluded view does not guarantee the successful observation of a frontier point as this can still be prevented by unknown occlusions or sensor noise.

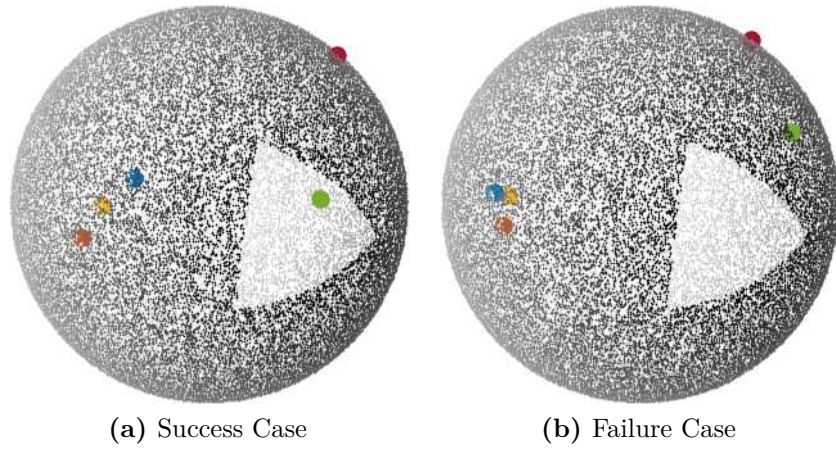


Figure 4.8: Experimental results demonstrating that the optimisation strategy (green dots) is not guaranteed to obtain an unoccluded view proposal if the initial solution, in this case an erroneous observing view (blue dots), does not lie in a region of free space on the sphere. (a) shows that an unoccluded view can sometimes be obtained but this is not guaranteed as the optimisation may converge to a local minima that is produced by a nonuniform distribution of points, as shown in (b).

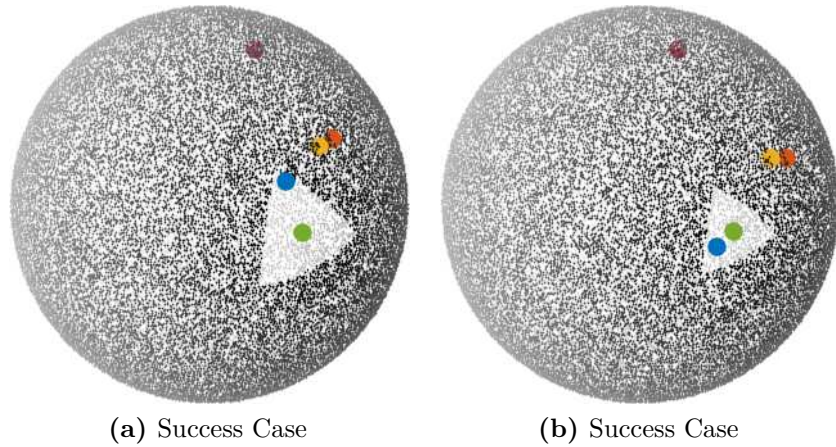


Figure 4.9: Experimental results demonstrating that the optimisation strategy (green dots) can obtain an unoccluded view proposal within small regions of free space on the sphere even for nonuniform distributions of projected points (a, b). This strategy is capable of proposing unoccluded views for occlusion configurations on which the other strategies fail as it directly identifies regions of free space on the sphere rather than evaluating the distribution of projected points. It is guaranteed to obtain a view free from known occlusions if the initial solution lies in free space (e.g., the observing view) but not for initial solutions in occupied regions (Fig. 4.8).

4.4 Evaluation

The presented strategies for proposing unoccluded views are evaluated by integrating them with SEE. Occlusion detection is performed using the adaptive search method (Sec. 4.2.3) for a given number of view proposals, τ , closest to the current sensor position. This is referred to as the view update limit. A new view is proposed using one of the strategies for any frontier point whose visibility from their current view proposal is found to be occluded. If the frontier point can not be successfully observed from this view (i.e., it is not reclassified as a core point) then the view is incrementally adjusted using the method from SEE (Sec. 3.2.5).

The performance of these strategies is compared with SEE by using the simulation environment discussed in Section 3.3. 100 experiments were performed on each of the standard models and the Radcliffe Camera model using the same simulated sensors (Table 3.2), algorithm parameters (Table 3.3) and performance metrics (Sec. 3.3.4). A view update limit of $\tau = 100$ views is used to detect occlusions within an occlusion search distance of $\psi = 1$ m for the standard models and $\psi = 40$ m for the Radcliffe Camera.

4.5 Discussion

The improvements in observation efficiency achievable by proactively handling occlusions before obtaining views are demonstrated by the experimental results (Fig. 4.10; Fig. 4.11; Table 4.1). All of the presented strategies for considering occlusion are able to obtain scene observations with an equivalent surface coverage to SEE using fewer views and shorter travel distances, except for the geodesic strategy on the Radcliffe Camera which travels a greater distance. The cost of this improved efficiency is typically an increase in computational time but this can be a valuable trade-off when it is necessary to account for the movement cost of a sensor platform.

Observation cost is typically reduced by requiring fewer views and shorter travel distances to observe a scene. Requiring fewer views means that less computational power is expended processing new measurements and shorter travel distances limit

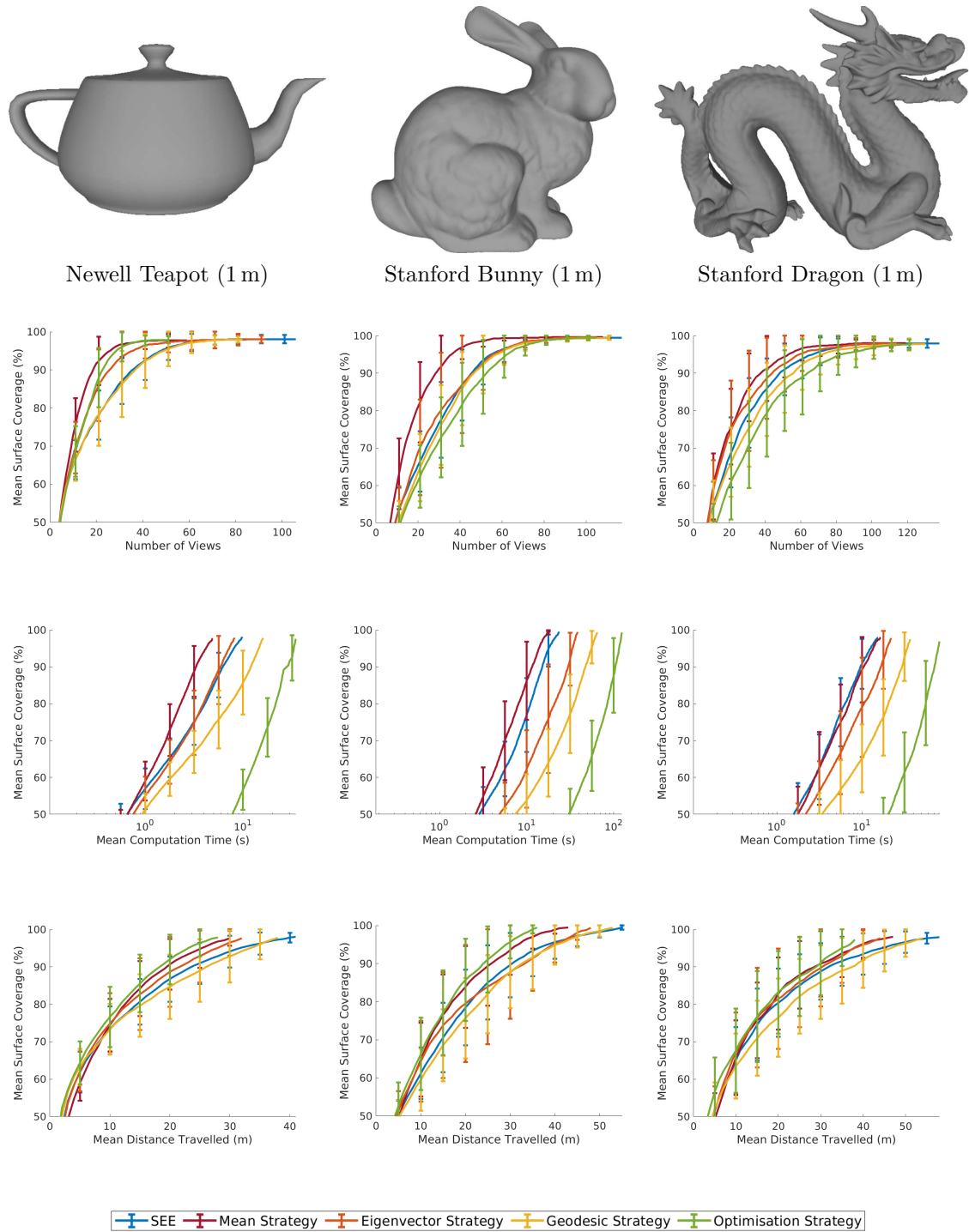


Figure 4.10: An experimental comparison of SEE with the presented occlusion strategies. The graphs show the mean surface coverage obtained by SEE and the presented strategies from 100 experiments relative to, from top to bottom, the number of views, the mean computation time and the mean travel distance.

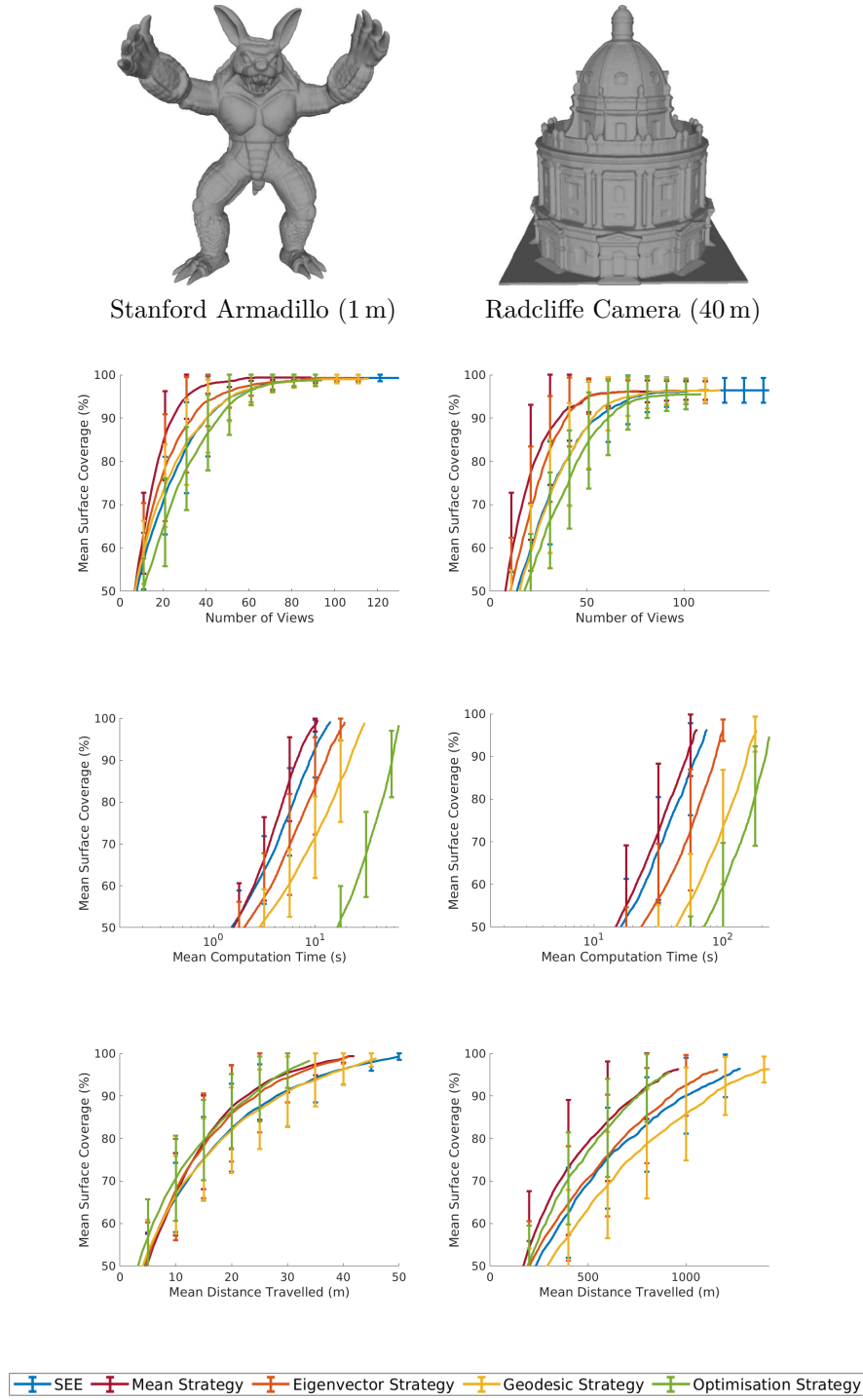


Figure 4.11: An experimental comparison of SEE with the presented occlusion strategies. The graphs show the mean surface coverage obtained by SEE and the presented strategies from 100 experiments relative to, from top to bottom, the number of views, the mean computation time and the mean travel distance.

	SEE	Mean	Eigenvector	Geodesic	Optimisation
Number of Views	56.9	28.3	35.8	49.5	31.9
Surface Coverage (%)	98.1	97.8	98.1	98.0	97.9
Computation Time (s)	9.91	4.94	8.24	16.1	35.0
Distance Travelled (m)	41.4	30.4	32.6	38.6	28.1

(a) Newell Teapot (Newell 1975)

	SEE	Mean	Eigenvector	Geodesic	Optimisation
Number of Views	73.7	48.1	60.6	68.4	67.6
Surface Coverage (%)	99.5	99.7	99.7	99.4	99.5
Computation Time (s)	23.9	18.5	38.6	64.7	125
Distance Travelled (m)	55.7	43.8	48.5	53.0	36.1

(b) Stanford Bunny (Turk and Levoy 1994)

	SEE	Mean	Eigenvector	Geodesic	Optimisation
Number of Views	79.2	62.7	57.5	72.2	73.1
Surface Coverage (%)	98.0	98.1	98.0	97.7	97.6
Computation Time (s)	15.6	16.7	22.0	37.1	82.0
Distance Travelled (m)	58.3	47.1	44.6	54.3	38.5

(c) Stanford Dragon (Curless and Levoy 1996)

	SEE	Mean	Eigenvector	Geodesic	Optimisation
Number of Views	65.3	42.6	52.3	58.3	60.1
Surface Coverage (%)	99.3	99.4	99.3	99.1	98.7
Computation Time (s)	14.3	10.8	19.7	30.7	67.2
Distance Travelled (m)	50.3	42.2	41.3	46.4	35.0

(d) Stanford Armadillo (Krishnamurthy and Levoy 1996)

	SEE	Mean	Eigenvector	Geodesic	Optimisation
Number of Views	63.5	41.1	43.2	61.1	62.7
Surface Coverage (%)	96.4	96.4	96.2	96.4	95.5
Computation Time (s)	75.5	63.2	100	184	232
Distance Travelled (m)	1280	965	1161	1425	917

(e) Radcliffe Camera (Boroczky 2016)

Table 4.1: The mean number of views captured, the mean surface coverage obtained, the mean computation time used and the mean travel distance required to observe four one-metre standard models (Newell Teapot, Stanford Bunny, Stanford Dragon, and Stanford Armadillo) and a 40 metre model of the Radcliffe Camera, calculated from 100 experiments with SEE and the presented occlusion strategies.

the energy consumed moving a sensor platform between views. The experimental results show that the greatest reductions in required views and travel distance are not achieved by the same presented strategies. The mean strategy obtains observations using the fewest views for all of the evaluated scenes except for the Stanford Dragon, on which the eigenvector strategy achieves the best view performance, but with greater travel distances than the optimisation strategy. The eigenvector strategy typically requires slightly more views than the mean strategy but uses a similar travel distance, except for on the Radcliffe Camera where it travels significantly farther.

The geodesic strategy does not achieve a significant improvement over SEE in terms of required views or travel distance but still incurs an increased computational cost. The optimisation strategy consistently obtains scene observations using the shortest travel distances but does not significantly reduce the number of views required, except for on the Newell Teapot, and incurs an increased computational cost.

In some cases the present strategies are able to achieve lower overall computation times than SEE by achieving a sufficiently significant reduction in the number of required views to mitigate the increased computational cost per view. The mean strategy is able to observe all of the scenes except for the Stanford Dragon using a lower mean computation time than SEE as it obtains observations using fewer views and only incurs a marginally greater computational cost per view. The eigenvector strategy observed the Newell Teapot using a lower computation time than SEE as it required 37% fewer views and did not incur a significant increase in the computational cost per view. Both the geodesic and optimisation strategies required greater computation times than SEE for all of the scenes as they did not obtain observations using much fewer views than SEE and majorly increased the computational cost per view due to their use of least-squares minimisation and nonlinear optimisation techniques. The computation time used by the optimisation strategy is reduced in subsequent work by combining it with methods to observe scenes using fewer views.

An explanation of the observation performances for the presented strategies is provided by an analysis, computed from the standard model experiments, of the mean number of views and travel distance required to successfully observe each

	SEE	Mean	Eigenvector	Geodesic	Optimisation
Number of Views	6.29	7.37	6.75	4.17	3.70
Distance Travelled (m)	2.68	2.17	2.16	1.88	1.13

Table 4.2: The mean number of views obtained and sensor distance travelled per frontier point for SEE and the strategies presented to propose unoccluded views. This analysis is calculated from the 100 experiments on each of the standard models.

frontier point (Table 4.2). The analysis shows that the geodesic and optimisation strategies successfully observe frontier points using fewer views than SEE while the mean and eigenvector strategies typically require a greater number of views. All of the presented strategies travel shorter distances to observe frontiers than SEE, with the optimisation strategy travelling the least. This demonstrates that the geodesic and optimisation strategies are the most successful at proposing unoccluded views.

The mean and eigenvector strategies are able obtain scene observations using fewer overall views than the geodesic and optimisation strategies despite being less successful at observing target frontier points (i.e., they require more views and greater travel distances to observe each frontier; Table 4.2). This is the result of the mean and eigenvector strategies failing to propose unoccluded views and instead utilising incremental view adjustments to obtain successful observations. These adjusted views are inadvertently able to attain large increases in scene coverage as multiple views are obtained while the sensor moves farther into unobserved scene regions. This means that these strategies can only observe scenes using fewer views by sacrificing the reliability of frontier observations for greater scene coverage per view.

In contrast, the geodesic and optimisation strategies are able to propose unoccluded views with greater reliability from which target frontier points can be successfully observed without requiring incremental views adjustments. This improvement in the efficiency of frontier observations means that fewer views and shorter travel distances are required to observe each frontier point (Table 4.2) but the increase in surface coverage obtained from each view is reduced as fewer measurements are captured from unobserved scene regions. Therefore more views are required to obtain complete observations. The optimisation strategy is able

to consistently observe scenes by travelling the shortest overall distances despite capturing more views as it achieves a significant reduction in the distance travelled per view but a similar improvement is not achieved by the geodesic strategy.

The objective of these strategies for proactively handling occlusions is to reliably propose unoccluded views from which target frontier points can be successfully observed. The optimisation strategy is shown to best achieve this objective as it observes frontier points with the greatest efficiency (Table 4.2) and is thus able to obtain scene observations using the shortest travel distances. This strategy does incur an increase in computation time but when observing scenes in the real world this would be offset by a reduction in the time required to move between views. A significant reduction in the number of views required to obtain scene observations is not achieved but this challenge is addressed independently in the following chapter by investigating methods for considering scene visibility when selecting next best views.

In summary, the work presented in this chapter makes three key contributions:

1. A computationally efficient method for detecting point-based occlusions.
2. A novel representation for encoding pointwise occlusion information.
3. An investigation of several strategies for proposing unoccluded views.

5

Considering Scene Visibility

Contents

5.1	Existing Methods	97
5.2	Constructing a Visibility Graph	100
5.2.1	Determining Frontier Visibility	100
5.2.2	Covering Visibility Graph	101
5.2.3	Defining Covisibility	103
5.2.4	Complete Covisibility Graph	104
5.3	Selecting a Next Best View	106
5.3.1	Global Minimum Covisibility	106
5.3.2	Global Maximum Visibility	108
5.3.3	Local Maximum Visibility	109
5.3.4	Local Maximum Visibility-Distance Ratio	110
5.4	Evaluation	112
5.5	Discussion	116

This chapter presents an investigation of methods for considering scene visibility when selecting next best views with an unstructured scene representation. The objective of this work is to increase the efficiency of scene observations by obtaining greater improvements in the coverage of scene surfaces from each captured view without requiring significantly longer travel distances between views. This can improve the observation performance of a NBV planning approach as it is possible to obtain highly complete coverage of scene surfaces using fewer views and less travelling.

The visibility of a scene from a set of proposed views can be quantified by the proportion of surfaces within the scene volume that are visible from at least one view. Visible surfaces are often observable from multiple view proposals so it is possible to obtain a complete scene observation by capturing a subset of views. This is achieved by selecting next best views with good visibility of unobserved and incompletely observed scene regions that can most improve the coverage of a scene observation.

The remainder of this chapter is organised as follows. Existing methods for evaluating scene visibility with structured approaches are reviewed in Section 5.1. Graphical representations for encoding the visibility of frontier points from proposed views, including a novel covisibility graph, are presented in Section 5.2. Next best view selection metrics that utilise the information encoded in this covisibility graph to consider the visibility of frontiers when selecting views are investigated in Section 5.3. The observation performance of these metrics is evaluated experimentally in comparison with SEE. The results are presented in Section 5.4 and discussed in Section 5.5.

The work presented in this chapter makes four key contributions:

1. An investigation of graphical representations for encoding information on the shared visibility of frontier points between proposed views.
2. The formulation of a novel covisibility graph for representing directed visibility relationships between individual views and frontier points.
3. An investigation of next best view selection metrics that utilise a graphical representation to consider frontier visibility when selecting views.
4. A next best view selection metric capable of selecting views that can provide significant improvements in surface coverage while travelling short distances.

5.1 Existing Methods

Scene visibility is quantified by determining the number, surface area or volume of represented manifolds within the viewing frustum of a sensor placed at a given view. NBV planning approaches with volumetric representations typically evaluate the number or cumulative entropy of voxels with a given state that are visible from a view. Approaches with surface representations often compute the coverage of a

triangulated mesh obtained from an initial observation by evaluating the surface area of the mesh that is visible from a set of proposed views. The visibility of manifolds in structured representations (e.g., voxels and triangulated surfaces) is typically evaluated using raycasting. As discussed in Chapter 4 raycasting is not suitable for assessing the visibility of zero-dimensional manifolds (i.e., points) in an unstructured representation. The methods used for evaluating manifold visibility therefore differ between these types of representation but similar metrics can be used to quantify scene visibility based on the number and mensuration of manifolds.

Volumetric approaches consider scene visibility by quantifying the unobserved space visible from each view. Most approaches with categorical voxel representations count the number of visible unobserved voxels (e.g., Connolly 1985; Massios and Fisher 1998; Banta et al. 2000; Vasquez-Gomez et al. 2014; Yoder and Scherer 2016; Monica and Aleotti 2018a). Some approaches restrict the counting to a subclass of unobserved voxels such as occplane voxels (Massios and Fisher 1998), occluded voxels (Banta et al. 2000) or frontier voxels (Yoder and Scherer 2016). Vasquez-Gomez et al. (2014) also count the number of occupied voxels to determine the overlap of a view with previous measurements. Approaches with a probabilistic voxel representation typically compute the cumulative entropy of visible voxels, as defined by their occupancy probability (e.g., Delmerico et al. 2018; Daudelin and Campbell 2017). Voxel visibility is also defined probabilistically for these approaches. Bircher et al. (2018) compute the cumulative volume of visible unobserved voxels. Selin et al. (2019) generalise the consideration of visibility to regions of unobserved space, defined using sparse raycasting, whose volume is computed by cubature integration.

Approaches with a surface representation quantify scene visibility based on the surface area of the triangulated mesh visible from a view (e.g., Reed and Allen 2000; Hollinger et al. 2012; Khalfaoui et al. 2013; Roberts et al. 2017; Peng and Isler 2019). Many multistage observation approaches aim to find a set of views that can provide the greatest improvement in the initial surface mesh given certain constraints and plan an observation trajectory to capture the measurements (e.g., Hollinger et al. 2012; Roberts et al. 2017; Peng and Isler 2019). Sufficient views are sampled to

provide complete coverage of the initial mesh and a subset with the best coverage is selected. View coverage can be quantified by the visible surface area, the angle of incidence between the view orientation and the mesh, a potential improvement in mesh density from new measurements or an estimated reduction in mesh uncertainty. The subset of views can be of fixed size or new views can be added until a coverage criterion is satisfied. Some approaches select next best views iteratively to observe the greatest area of surfaces whose visibility was occluded, either partially or fully, from previous views (e.g., Reed and Allen 2000; Khalfaoui et al. 2013).

Approaches using a combination of volumetric and surface representations typically consider scene visibility by quantifying both the unobserved space and mesh surface area visible from each view. Kriegel et al. (2015) use a weighted sum of the mean entropy for all visible voxels and a mesh quality metric. The mesh quality is evaluated as a weighted sum of the average mesh density and the percentage of boundary edges in the visible region of the mesh. Song and Jo (2018) consider the visibility of frontier voxels and points extracted from a surface mesh. Sufficient views are sampled to provide coverage of every surface point and a subset of views is selected that still maintain complete coverage. An observation trajectory is planned through the subset of views in order to provide coverage of all frontier voxels within a given distance of the path. Monica and Aleotti (2018b) present a hybrid surfel representation which quantifies visibility based on the cumulative surface area of visible frontels (i.e., surfels at the boundary of observed and unobserved space).

The investigation of methods for considering scene visibility with an unstructured representation presented in this chapter does not directly utilise techniques from structured representations but applies manifold counting (i.e., the number of frontier points), as used by many volumetric approaches, and considers the coverage of represented manifolds (i.e., the visibility of frontiers) from a set of view proposals, similar to the mesh coverage computed by many approaches with a surface representation.

5.2 Constructing a Visibility Graph

The improvement in scene coverage obtainable from a view proposal can be quantified by considering the number of visible frontier points. The visibility of a frontier point from a given view is determined using the occlusion detection approach presented in the previous chapter (Sec. 5.2.1). The visibility of frontier points from the set of proposed views is encoded in a graphical representation, $\mathcal{G} = (V, E)$, where view proposals are represented by the set of vertices, V , and the edges between them, E , denote the visibility relationships that are considered when selecting a next best view.

This section presents an investigation of two graphical representations for encoding the visibility of frontier points: a covering visibility graph and a complete covisibility graph. The covering visibility graph determines the visibility of each frontier point from every proposed view and computes a minimal subset of view proposals sufficient to obtain an observation of every frontier. In this graph the subset of views are represented by vertices and fully connected with undirected edges denoting the travel distance between every pair of views (Sec. 5.2.2).

The complete covisibility graph is a novel representation that encodes information on the shared visibility of each frontier point from multiple view proposals. In this graph every view proposal is represented by a vertex in the graph and directed edges denote that the frontier point associated with the child vertex of an edge is *covisible* (Sec. 5.2.3) from the view proposal associated with the parent vertex (Sec. 5.2.4).

5.2.1 Determining Frontier Visibility

The visibility of a frontier point from a proposed view is determined using the adaptive search method for detecting occlusions (Sec. 4.2.3). This searches the sight line between a view and frontier for occluding points. It provides a sufficient determination of visibility when the view distance is large enough that a significant proportion of the scene volume lies within the viewing frustum of the sensor.

If a scene is observed at a relatively short view distance it may also be desirable to ensure that the sight line is within the viewing frustum of the sensor. This is guaranteed when determining the visibility of a frontier point from its associated view

as the view is proposed with a sight line intersecting the frontier. It is not known to be true when evaluating the visibility of a frontier point from a different view proposal.

The frustum visibility of a frontier can be computed exactly if the full sensor pose is known by defining a frustum from the sensor field-of-view and computing its intersection with the sight line. If the full sensor pose is not known then the viewing frustum can be approximated by a viewing cone with its apex at the view position and an axis defined by the view orientation. In this case a frontier point is considered visible if the angle between the view orientation, ϕ_j , and the reverse of the sight line, $\mathbf{w}_{kj}$, from the frontier point, $\mathbf{f}_k$, to the view position, $\mathbf{x}_j$, is less than angle of the viewing cone, as defined by the minimum sensor field-of-view angle,

$$\arccos\left(-\frac{\mathbf{w}_{kj} \cdot \phi_j}{\|\mathbf{w}_{kj}\| \|\phi_j\|}\right) < \theta_{\min}, \quad (5.1)$$

where $\theta_{\min}$ is the minimum of horizontal, θ_x , and vertical, θ_y , field-of-view angles.

This consideration of frustum visibility was investigated but ultimately not included as SEE represents views using position and orientation vectors rather specifying a complete sensor pose. The viewing cone approximation was evaluated but found to be too restrictive for sensors with an oblong viewing frustum (e.g., a LiDAR with $\theta_x = 90^\circ$ and $\theta_y = 30^\circ$). The detection of occluding points can provide a sufficiently robust evaluation of frontier visibility as it is capable of identifying the restricted visibility of sight lines that have an acute angle with scene surfaces.

5.2.2 Covering Visibility Graph

The covering visibility graph is used to represent a sufficient subset of view proposals to provide visibility of every frontier point (Fig. 5.1). The graph is fully connected with undirected edges denoting the travel distance between views. This representation is best suited for planning a trajectory of next best views that can obtain an observation of every frontier point or observe the greatest number of frontiers while travelling the shortest distance. A suitable view trajectory can be found by computing a solution to the Travelling Salesman Problem (TSP). Similar approaches are presented by Song and Jo (2017) and Song and Jo (2018).

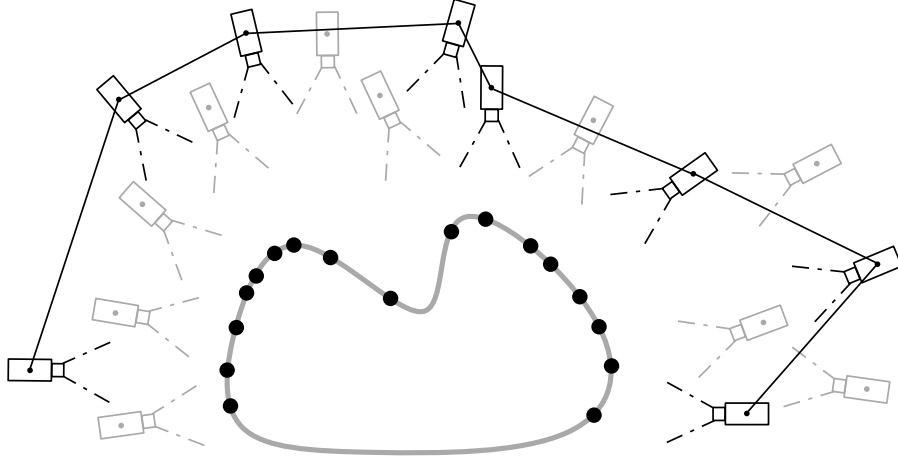


Figure 5.1: An illustration of the covering visibility graph representation. The chosen set of view proposals from which all frontier points (black dots) are visible is shown as a set of sensors in black. Unselected view proposals are shown in grey. The black line connecting the chosen view proposals represents a view trajectory.

The subset of view proposals used to construct the covering visibility graph is found by computing an approximate solution to the minimum covering set problem (Karp 1972). This can be formally expressed as a covering set for the set of all view proposals, $W' \subseteq W$, which satisfies the constraint that every frontier in the set of all frontiers, F , must be visible from at least one view,

$$W' \subseteq W \quad \text{s.t.} \quad F \equiv \bigcup_{\mathbf{v} \in W'} F_{\mathbf{v}}, \quad (5.2)$$

where $F_{\mathbf{v}}$ is the set of frontiers visible from a given view, $\mathbf{v}$.

Karp (1972) shows that finding the minimum covering set is an NP-complete problem and therefore it is only possible to find an approximate solution in polynomial time. An approximate solution can be found using a greedy algorithm (Chvatal 1979). At each iteration a view is selected with the greatest number of visible frontier points that are not yet covered. New views are chosen until coverage of every frontier point is attained. Chvatal (1979) shows that the covering set found has a cardinality no larger than $\log(|W|)$ times that of the minimal set.

Views in the covering set are represented by vertices in an undirected graph. The travel distance between every pair of views is associated with an edge connecting the corresponding vertices, making the graph fully connected. A view trajectory can be

planned using this connectivity with the aim of obtaining the greatest improvement in scene coverage while travelling the shortest distance. This trajectory can be unconstrained in order to improve global scene coverage or constrained by a limit on the number of visited view proposals and travel distance.

The use of a trajectory planning method to select a sequence of next best views demonstrated that significant increases in global scene coverage could be obtained at the expense of producing numerous local regions of partially observed surfaces. This was the result of obtaining new point measurements, some classified as frontier points, from unobserved scene surfaces that were not considered until the covering graph was updated to include their view proposals and a new view trajectory was planned. These new trajectories were typically required to re-traverse a similar path in order to fully observe the new frontier points, increasing the overall travel distance.

The coverage of local scene regions was improved by using constrained view trajectories (i.e., limited by a number of views or travel distance) but this incurred an increase in computational time due to the complexity of performing frequent updates to the minimum covering set of vertices and the graph connectivity. Using shorter trajectories also reduced the obtainable improvement in scene coverage.

Therefore, while a covering visibility graph may be a suitable representation for NBV planning approaches that obtain measurements from a single view trajectory and then process them offline (e.g., many surface-based approaches), it is not a suitable choice for a NBV planning approach that performs incremental measurement updates. This motivated the formulation of a covisibility graph that can be updated more efficiently and encode detailed information on local scene coverage.

5.2.3 Defining Covisibility

Covisibility is a novel concept that refers to the shared visibility of a given frontier point from multiple view proposals. A covisibility relationship exists between two proposed views, $\mathbf{v}_i$ and $\mathbf{v}_j$, if the frontier point associated with one of the views, $\mathbf{f}_i$ or $\mathbf{f}_j$, is visible from the other view. The covisibility between views is encoded in a graphical representation using directed edges (Fig. 5.2). A directed edge from a

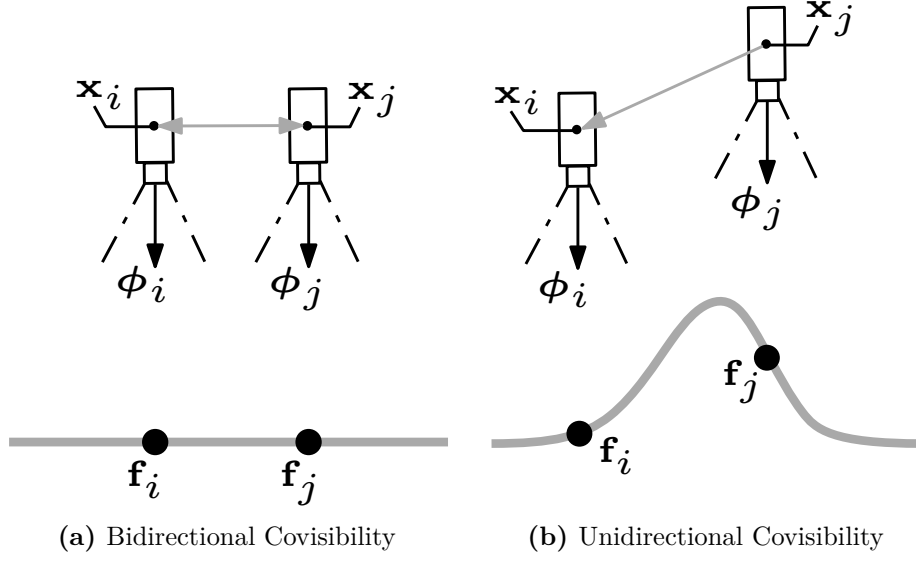


Figure 5.2: Illustrations of the different covisibility relationships (grey arrows) that can exist between two views (black sensors), $\mathbf{v}_i = \{\mathbf{x}_i, \phi_i\}$ and $\mathbf{v}_j = \{\mathbf{x}_j, \phi_j\}$, for their associated frontier points (black dots), $\mathbf{f}_i$ and $\mathbf{f}_j$. (a) shows two views that share a bidirectional covisibility relationship as the frontier point $\mathbf{f}_i$ is visible from the view $\mathbf{v}_j$ and an equivalent relationship exists between the view $\mathbf{v}_i$ and the frontier point $\mathbf{f}_j$. (b) illustrates a unidirectional covisibility relationship between two views in which the frontier point $\mathbf{f}_i$ is visible from the view $\mathbf{v}_j$ but the frontier point $\mathbf{f}_j$ is not visible from the view $\mathbf{v}_i$ due to the presence of an occluding surface.

parent vertex, $\mathbf{v}_i$, to a child vertex, $\mathbf{v}_j$, denotes that the frontier point associated with the child vertex, $\mathbf{f}_j$, is covisible from the view associated with the parent vertex. The covisibility relationship between two views can be *bidirectional* if each of the associated frontier points is visible from both views (Fig. 5.2a) or *unidirectional* if only one of the views has unoccluded visibility of both frontier points (Fig. 5.2b).

5.2.4 Complete Covisibility Graph

The complete covisibility graph represents the set of all view proposals and their shared visibility of frontier points (Fig. 5.3). It encodes detailed information on the number of frontiers visible from each proposed view and the covisibility of frontier points between views. This makes it easy to identify which view proposals can provide the best local improvement in scene coverage while incurring short

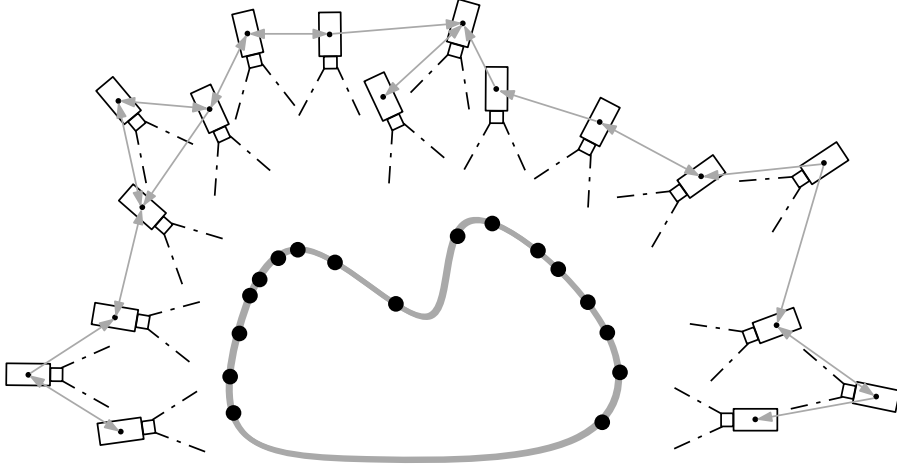


Figure 5.3: An illustration of the complete covisibility graph representation. The connectivity between a set of view proposals (black sensors) is represented with directed edges (grey arrows). An edge from a parent view to a child view denotes that the frontier (black dots) associated with the child is visible from the parent.

travel distances. The graph can be efficiently updated when new measurements are obtained as only local computations are performed when adding and removing views.

The complete graph is constructed by determining the covisibility between views of the frontiers they were proposed to observe. Every frontier point is associated with the view proposed to observe it. This association is represented by a frontier-view pair, $\mathbf{m} = \{\mathbf{v}, \mathbf{f}\}$, denoting that the view, $\mathbf{v}$, is proposed to observe the frontier point, $\mathbf{f}$. The complete covisibility graph is a directed graph, $\mathcal{G} = (M, E)$, which represents the shared visibility between these pairs. Vertices in the graph represent frontier-view pairs. An edge, $(\mathbf{m}_j, \mathbf{m}_k) \in E$, exists from a parent vertex, $\mathbf{m}_j$, to a child vertex, $\mathbf{m}_k$, if the child frontier point, $\mathbf{f}_k$, is visible from the parent view, $\mathbf{v}_j$,

$$D'(\mathbf{f}_k, \mathbf{v}_j) \equiv \emptyset \implies (\mathbf{m}_j, \mathbf{m}_k) \in E, \quad (5.3)$$

where $D'(\mathbf{f}, \mathbf{v})$ is the set of occluding points found by the adaptive search as in (4.4).

The graph is updated when a new set of sensor measurements is obtained and after all point classifications and view updates have been performed. Vertices associated with frontiers that have been reclassified are removed and vertices are added to represent new frontier-view pairs. The graph connectivity is updated by removing all edges associated with a parent or child vertex that no longer exists.

New edges are added by evaluating the visibility of new frontier points from existing view proposals and the visibility of all frontier points from new view proposals.

The complete covisibility graph provides a more detailed representation of the shared visibility between view proposals than a covering graph. Every view proposal is represented rather than only a covering subset and each edge denotes the visibility of a specific frontier point from a given view in addition to the required travel distance between the proposed views. The number of view proposals with visibility of a given frontier point is denoted by the indegree (i.e., the number of incoming edges), $\deg^-(\mathbf{m})$, of the vertex associated with its frontier-view pair. The number of frontiers that are visible from a view is given by the outdegree (i.e., the number of outgoing edges), $\deg^+(\mathbf{m})$, of the vertex corresponding to the frontier-view pair.

This information is used to select next best views by considering the number of frontiers visible from a view and the set of view proposals from which a given frontier point is visible. Views can be chosen to observe a specific frontier point while also providing the greatest coverage of other frontiers. Metrics for selecting next best views using a complete covisibility graph are discussed in the following section.

5.3 Selecting a Next Best View

Next best views are selected to provide the greatest improvement in scene coverage within certain constraints (e.g., travel distance or the visibility of a specific frontier point). This section presents different metrics for selecting next best views using a complete covisibility graph. These metrics aim to improve the efficiency of scene observations by selecting views to observe frontier points with poor visibility or maximise the number of visible frontiers while reducing the required travel distance.

5.3.1 Global Minimum Covisibility

The complete covisibility graph encodes information on which view proposals can observe each frontier point. Frontiers that are visible from fewer views are more likely to lie on occluded surfaces. Improving the coverage of a scene observation depends on obtaining sensor measurements from these occluded surfaces. It can be

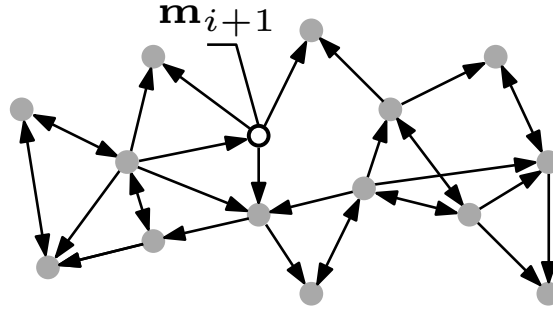


Figure 5.4: An illustration of the Global Minimum Covisibility (GMC) metric for selecting next best views. Vertices (grey dots) in the complete covisibility graph are connected with edges denoting visibility (black arrows). The next best view is associated with the vertex (black circle), $\mathbf{m}_{i+1}$, that has the smallest indegree.

advantageous to prioritise the observation of frontier points with limited covisibility (i.e., few observing views) to ensure that poorly observed regions (i.e., those with low measurement density) are fully observed before the scene observation is extended into unobserved space. The selection of views to observe frontiers with limited covisibility is prioritised by considering the indegree of vertices in the complete covisibility graph.

The Global Minimum Covisibility (GMC) metric selects a next best view, $\mathbf{v}_{i+1}$, to observe the frontier point that is visible from the least number of views (Fig. 5.4). This is achieved by selecting the view proposal corresponding to a vertex in the graph with lowest indegree (i.e., the fewest incoming edges),

$$\mathbf{m}_{i+1} = \arg \min_{\mathbf{m} \in M} \left(\deg^{-}(\mathbf{m}) \right) . \quad (5.4)$$

If multiple vertices have the same minimum number of incoming edges then the next best view is the proposed view in this subset closest to the current sensor position.

Selecting next best views using the GMC metric prioritises improving the coverage of poorly observed scene regions over selecting views that may obtain a greater increase in coverage by extending the scene observation into unobserved space. This can be advantageous when obtaining a scene observation using a system with strict time or energy constraints that may preclude the capture of a complete observation. In this scenario it may be preferable to obtain complete coverage of one scene region rather than partial coverage of the entire scene. However, when observing scenes with an unconstrained system it is usually better to prioritise

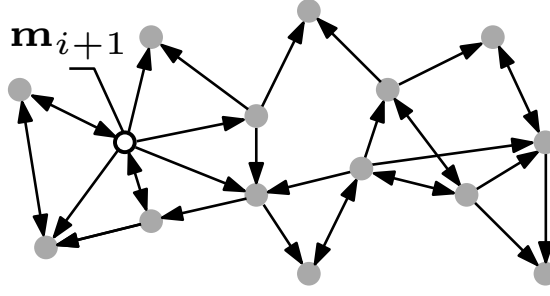


Figure 5.5: An illustration of the Global Maximum Visibility (GMV) metric for selecting next best views. Vertices (grey dots) in the complete covisibility graph are connected with edges denoting visibility (black arrows). The next best view is associated with the vertex (black circle), $\mathbf{m}_{i+1}$, that has the greatest outdegree.

obtaining the best improvement in scene coverage from every view as this typically provides a greater reduction in the number of views required to obtain an observation.

5.3.2 Global Maximum Visibility

Obtaining a scene observation using the fewest number of views is typically achieved by greedily selecting next best views with the greatest number of visible frontier points. In a complete covisibility graph the number of frontiers visible from a proposed view is given by the number of outgoing edges from its associated vertex.

The Global Maximum Visibility (GMV) metric selects a next best view, $\mathbf{v}_{i+1}$, to observe the greatest number of frontier points (Fig. 5.5). This is achieved by selecting the view proposal associated with a vertex in the graph that has the greatest outdegree (i.e., the most outgoing edges),

$$\mathbf{m}_{i+1} = \arg \max_{\mathbf{m} \in M} (\deg^+(\mathbf{m})) . \quad (5.5)$$

If multiple vertices have the same maximum number of outgoing edges then the next best view is the proposed view in this subset closest to the current sensor position.

Next best views selected with the GMV metric usually provide the greatest improvement in scene coverage and allow an observation to be completed using the fewest number of views. The cost of this improved observation performance is often an increase in sensor travel distance between views as there is no constraint on how far the sensor can travel to obtain a view with the most visible frontier points. The

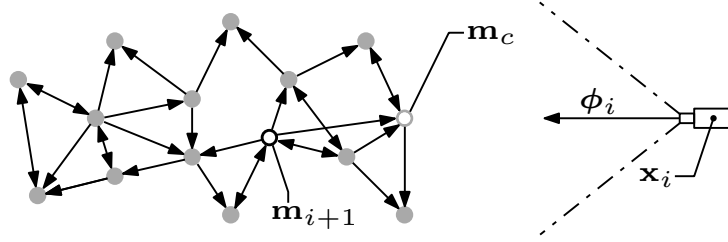


Figure 5.6: An illustration of the Local Maximum Visibility (LMV) metric for selecting next best views. Vertices (grey dots) in the frontier visibility graph are connected with edges denoting visibility (black arrows). The sensor represents the current view, $\mathbf{v}_i = \{\mathbf{x}_i, \phi_i\}$. The next best view is associated with the vertex (black circle), $\mathbf{m}_{i+1}$, that has the greatest outdegree and can observe the frontier associated with the vertex (grey circle), $\mathbf{m}_c$, whose view is closest to the sensor position.

overall travel distance can be lower than when using other view selection metrics if the reduction in the number of views obtained is significant enough to outweigh the increased travel distance between the views, but this is not guaranteed.

5.3.3 Local Maximum Visibility

The travel distance between views can be implicitly constrained by requiring the visibility of frontier points associated with views close to the current sensor position. This prioritises the selection of next best views whose viewing frustums overlap with the current view frustum and prevents the sensor from moving to capture views of a distant unobserved scene region instead of obtaining new measurements from nearby regions that are poorly observed. A suitable choice of frontier for constraining the sensor travel distance is the point associated with the view proposal closest to the current sensor position. The best possible improvement in scene coverage can be obtained while satisfying this constraint by selecting a next best view with the maximum number of visible frontier points from the set of permissible views.

The Local Maximum Visibility (LMV) metric selects a next best view, $\mathbf{v}_{i+1}$, to observe the greatest number of frontier points while requiring the visibility of the frontier point associated with the vertex whose view proposal, $\mathbf{m}_c$, is closest to the current view (Fig. 5.6). This is achieved by selecting the next best view from a vertex set, M_c , containing the closest view proposal and the parent vertices

of incoming edges to the vertex,

$$M_c := \{\mathbf{m}_c\} \cup \{\mathbf{m} \in M \mid (\mathbf{m}, \mathbf{m}_c) \in E\}, \quad (5.6)$$

where $\mathbf{v}_i = \{\mathbf{x}_i, \phi_i\}$ denotes the current view and

$$\mathbf{m}_c = \arg \min_{\mathbf{m} \in M} (||\mathbf{x} - \mathbf{x}_i||). \quad (5.7)$$

The next best view selected from this set has the greatest number of outgoing edges,

$$\mathbf{m}_{i+1} = \arg \max_{\mathbf{m} \in M_c} (\deg^+(\mathbf{m})). \quad (5.8)$$

If multiple vertices have the same maximum number of outgoing edges then the view proposal in this subset closest to the current sensor position is the next best view.

Selecting next best views using the LMV metric reduces the travel distance required per view when compared with the GMV metric that chooses views to maximise the global improvement in scene coverage. The observation of poorly observed scene regions close to the current sensor position is prioritised and a next best view is selected to provide the best improvement in scene coverage possible within the constrained set of permissible views. The reduction in travel distance provided by using this metric is limited as the frontier visibility requirement only implicitly constrains the travel distance. For example, when choosing between a view, $\mathbf{v}_i$, at a given distance, x , from which n frontier points are visible and a view, $\mathbf{v}_j$, at a distance, $2x$, from which $n + 1$ frontiers are visible, the second view will be chosen even though it requires twice the travel distance to obtain a marginally greater improvement in scene coverage. A more efficient metric would explicitly consider the travel cost associated with observing each frontier when selecting a next best view.

5.3.4 Local Maximum Visibility-Distance Ratio

The Local Maximum Visibility-Distance Ratio (LMR) metric still prioritises the selection of views that can provide the best improvement in scene coverage but also considers the diminishing return of moving farther than necessary. An observation value for views which explicitly considers the sensor travel distance is quantified by

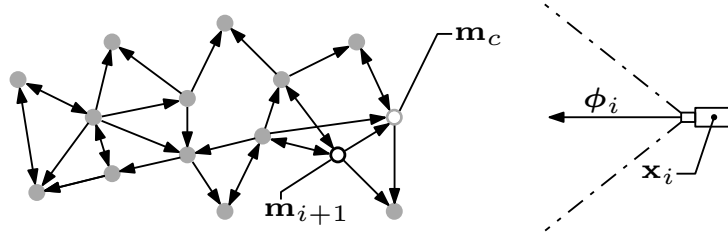


Figure 5.7: An illustration of the Local Maximum Visibility-Distance Ratio (LMR) metric for selecting next best views. Vertices (grey dots) in the frontier visibility graph are connected with edges denoting visibility (black arrows). The sensor represents the current view, $\mathbf{v}_i = \{\mathbf{x}_i, \phi_i\}$. The next best view is the view associated with the vertex (black circle), $\mathbf{m}_{i+1}$, that has the greatest outdegree relative to its distance from the sensor position, $\mathbf{x}_i$, and can observe the frontier associated with the vertex (grey circle), $\mathbf{m}_c$, whose view is closest to the sensor.

a ratio between the number of visible frontier points and the required travel distance. This evaluation penalises the relative value of views that are far away from the current sensor position but are not likely to provide a significantly greater improvement in the scene observation than closer views with fewer visible frontier points.

A next best view, $\mathbf{v}_{i+1}$, is selected to observe the greatest number of frontier points while travelling the shortest distance from the current view (Fig. 5.7). The frontier point associated with the vertex having the closest view proposal, $\mathbf{m}_c$, to the current view, $\mathbf{v}_i = \{\mathbf{x}_i, \phi_i\}$, is required to be visible from the selected view, as discussed in Section 5.3.3, and the same vertex set containing $\mathbf{m}_c$ and parent vertices of its incoming edges is computed, as in (5.6). The next best view is the view proposal in this set with the greatest number of outgoing edges relative to the required travel distance,

$$\mathbf{m}_{i+1} = \arg \max_{\mathbf{m} \in M_c} \left(\frac{\deg^+(\mathbf{m})}{\|\mathbf{x} - \mathbf{x}_i\|} \right). \quad (5.9)$$

This equation produces a ratio between the number of visible frontiers and required travel distance that is often significantly greater for the view proposal closest to the current sensor position than other permissible views. The result is that in most circumstances the closest view proposal is selected as the next best view, limiting the potential improvement in scene coverage. This bias is overcome by the Exclusionary Local Maximum Visibility-Distance Ratio (EMR) metric, which

excludes the closest view proposal from the set of permissible vertices and only considers view proposals that can observe a greater number of frontier points than $\mathbf{m}_c$ (i.e., vertices with a greater number of outgoing edges),

$$M'_c := \{\mathbf{m} \in M \mid (\mathbf{m}, \mathbf{m}_c) \in E \wedge \deg^+(\mathbf{m}) > \deg^+(\mathbf{m}_c)\}. \quad (5.10)$$

If none of the evaluated view proposals have a greater outdegree than $\mathbf{m}_c$, $M'_c \equiv \emptyset$, then the next best view is chosen to be the closest view proposal, $\mathbf{m}_{i+1} = \mathbf{m}_c$.

The LMR metric for selecting next best views directly considers the trade-off between observation cost, as quantified by the travel distance, and performance, as given by the number of visible frontier points. The overall efficiency of scene observations is improved by reducing both the travel distance and number of views required to obtain scene observations. The LMR metric formulation resulted in the closest view proposal being weighted significantly higher than other views but this limitation was overcome by creating the EMR metric. This only considers permissible views with better frontier visibility than the closest view proposal.

5.4 Evaluation

The presented metrics for selecting next best views with a complete covisibility graph are evaluated by substituting the view selection method used by SEE with an implementation of each metric. Updating the complete covisibility graph becomes computationally expensive when processing changes in connectivity for a large number of frontier points. This can occur when a significant quantity of new frontiers are identified after obtaining measurements from a sizeable scene region that was previously unobserved. The resulting increase in computational cost is constrained by using the view update limit (Sec. 4.4) to restrict connectivity updates for the graph to the $\tau = 100$ vertices with view proposals closest to the sensor position.

The performance of these metrics is compared with SEE by using the simulation environment from Section 3.3. 100 experiments were performed on each of the standard models and the Radcliffe Camera using the same simulated sensors (Table 3.2), algorithm parameters (Table 3.3) and performance metrics (Sec. 3.3.4).

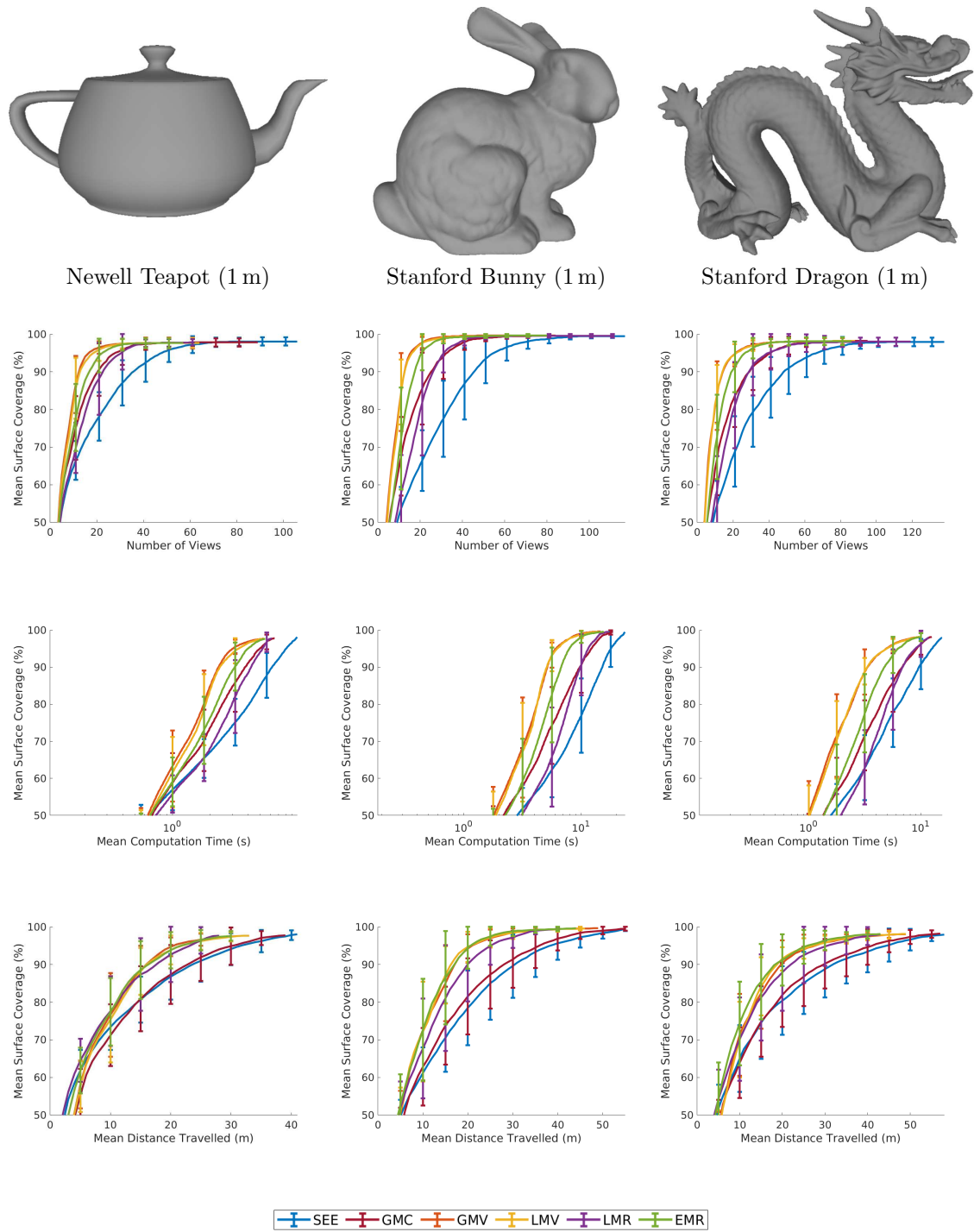


Figure 5.8: An experimental comparison of SEE with the presented next best view selection metrics. The graphs show the mean surface coverage obtained by SEE and the presented metrics from 100 experiments relative to, from top to bottom, the number of views, the mean computation time and the mean travel distance.

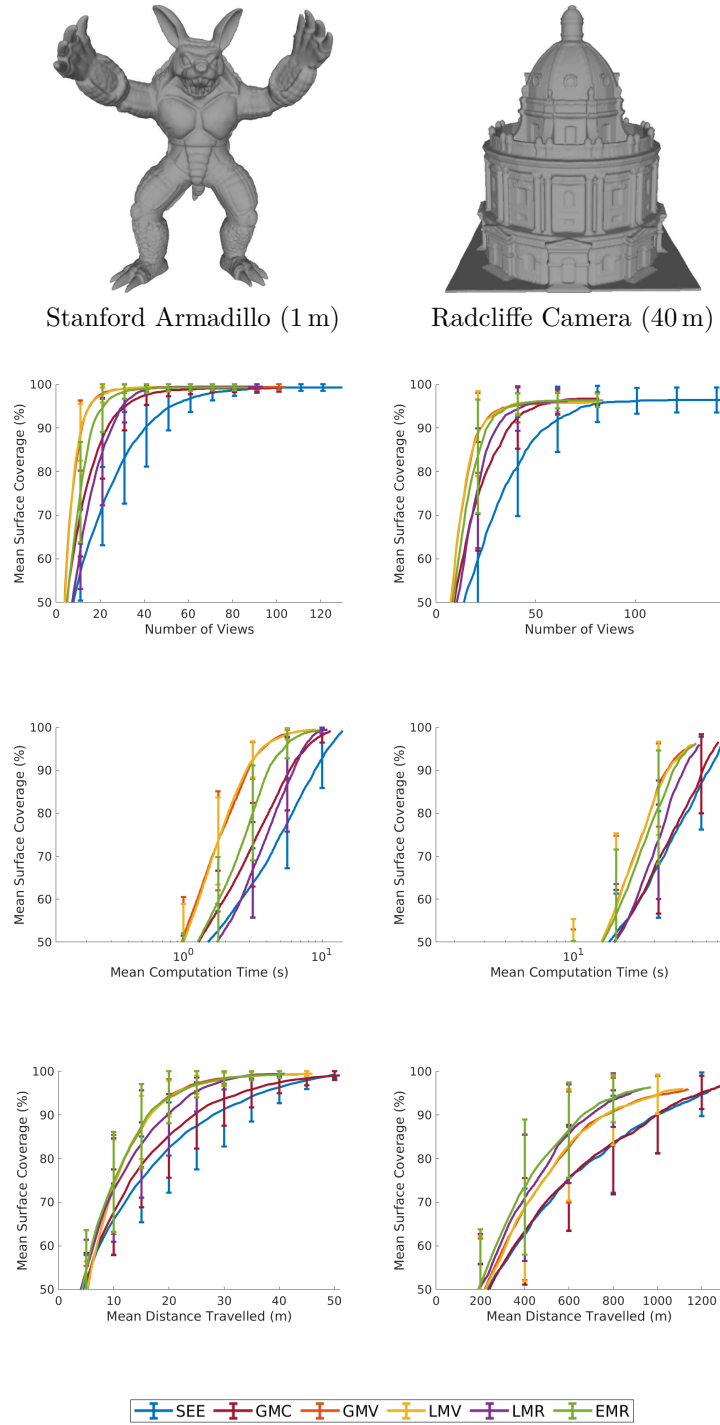


Figure 5.9: An experimental comparison of SEE with the presented next best view selection metrics. The graphs show the mean surface coverage obtained by SEE and the presented metrics from 100 experiments relative to, from top to bottom, the number of views, the mean computation time and the mean travel distance.

	SEE	GMC	GMV	LMV	LMR	EMR
Number of Views	56.9	37.1	30.4	30.8	34.3	30.8
Surface Coverage (%)	98.1	97.8	97.7	97.7	97.8	97.6
Computation Time (s)	9.91	6.58	5.33	5.41	6.23	5.53
Distance Travelled (m)	41.4	39.5	33.1	33.5	28.7	30.5
(a) Newell Teapot (Newell 1975)						
	SEE	GMC	GMV	LMV	LMR	EMR
Number of Views	73.7	53.2	42.4	41.1	49.2	45.5
Surface Coverage (%)	99.5	99.6	99.6	99.5	99.5	99.6
Computation Time (s)	23.9	18.8	14.7	14.1	17.1	15.7
Distance Travelled (m)	55.7	55.9	49.5	46.0	42.5	44.0
(b) Stanford Bunny (Turk and Levoy 1994)						
	SEE	GMC	GMV	LMV	LMR	EMR
Number of Views	79.2	56.8	43.2	45.7	52.2	45.3
Surface Coverage (%)	98.0	98.2	98.1	98.2	98.0	98.1
Computation Time (s)	15.6	12.5	9.33	9.77	11.8	10.1
Distance Travelled (m)	58.3	57.2	49.1	49.9	43.4	43.7
(c) Stanford Dragon (Curless and Levoy 1996)						
	SEE	GMC	GMV	LMV	LMR	EMR
Number of Views	65.3	49.6	38.9	40.9	47.6	41.5
Surface Coverage (%)	99.3	99.2	99.3	99.5	99.4	99.3
Computation Time (s)	14.3	11.7	8.91	9.10	11.1	9.58
Distance Travelled (m)	50.3	51.2	44.8	46.7	41.6	41.1
(d) Stanford Armadillo (Krishnamurthy and Levoy 1996)						
	SEE	GMC	GMV	LMV	LMR	EMR
Number of Views	63.5	44.5	34.1	34.1	37.6	35.7
Surface Coverage (%)	96.4	96.7	95.9	96.0	96.1	96.4
Computation Time (s)	75.5	71.2	49.7	50.1	55.3	53.0
Distance Travelled (m)	1280	1294	1139	1118	941	971
(e) Radcliffe Camera (Boroczyk 2016)						

Table 5.1: The mean number of views captured, the mean surface coverage obtained, the mean computation time used and the mean travel distance required to observe four one-metre standard models (Newell Teapot, Stanford Bunny, Stanford Dragon, and Stanford Armadillo) and a 40 metre model of the Radcliffe Camera, calculated from 100 experiments with SEE and the presented view selection metrics.

	SEE	GMC	GMV	LMV	LMR	EMR
Surface Coverage (%)	1.44	2.01	2.55	2.49	2.15	2.42
Distance Travelled (m)	0.75	1.04	1.14	1.11	0.85	0.98

Table 5.2: The mean surface coverage attained and distance travelled per view for SEE and the evaluated metrics, calculated from the standard model experiments.

5.5 Discussion

The experimental results (Fig. 5.8; Fig. 5.9; Table 5.1) demonstrate the value of selecting next best views using metrics that consider scene visibility. All of the evaluated metrics outperform SEE, which uses a view selection metric that considers sensor travel distance but not scene visibility, in terms of the computational cost and number of views required to obtain observations for every model. The difference in the mean number of views is particularly significant for the full-scale model of the Radcliffe Camera as integrating any of the present metrics, except for GMC, with SEE reduces the mean number of views required by more than 40% (Table 5.1e). This can be attributed to the significant magnitude of visibility changes that occur when moving a sensor at a far distance to observe a scene with large surface areas as this increases the improvement gained by considering scene visibility.

Every metric except for GMC obtains scene observations using a lower mean travel distance than SEE. The GMC metric uses a greater travel distance than SEE when obtaining observations of the Stanford Bunny, Stanford Armadillo and Radcliffe Camera. Its travel distance is only marginally lower than SEE for the Newell Teapot and Stanford Dragon. As the GMC metric is formulated to prioritise the observation of poorly observed regions over extending coverage of the scene it often obtains a lower increase in surface coverage per view than the other metrics (Table 5.2) and therefore requires more views to observe a scene. This produces a greater overall travel distance as the increase in travel distance per view does not correspond with a sufficient decrease in the number of views obtained.

The relative performance of the evaluated metrics was compared to determine which provided the best overall improvement in observation efficiency and would

be integrated into subsequent work on SEE. The statistical analysis presented in Table 5.2 shows that no metric outperforms the others in terms of both surface coverage and travel distance. The GMV metric, which selects a next best view with the global maximum number of visible frontiers, obtains the greatest surface coverage per view. The metric used by SEE, which selects views to reduce the sensor travel distance, has the lowest travel distance per view. The metric that provides the best overall improvement in observation performance when compared to SEE is identified by considering the percentage improvement in surface coverage relative to the percentage increase in distance. The EMR metric provides the greatest improvement in observation performance as it increases the surface coverage per view by 68% compared with SEE while only requiring a 31% increase in travel distance.

The observation performance of SEE is shown to be significantly improved by considering scene visibility when selecting a next best view. The complete covisibility graph encodes detailed information on the shared visibility of frontier points between view proposals in an efficient representation. The investigation of different metrics for selecting a next best view using this representation demonstrates the value of considering scene visibility when selecting next best views. The metric with the greatest observation performance, EMR, considers a ratio between the number of visible frontier points and required travel distance. This metric and the optimisation strategy for proactively handling occlusions (Ch. 4) are integrated with SEE to create SEE++, an improved version of SEE that is presented in the following chapter.

In summary, the work presented in this chapter makes four key contributions:

1. An investigation of graphical representations for encoding information on the shared visibility of frontier points between proposed views.
2. The formulation of a novel covisibility graph for representing directed visibility relationships between individual views and frontier points.
3. An investigation of next best view selection metrics that utilise a graphical representation to consider frontier visibility when selecting views.
4. A next best view selection metric capable of selecting views that can provide significant improvements in surface coverage while travelling short distances.

6

Observing Scenes with Fewer Views and Less Travelling

Contents

6.1	SEE++	120
6.1.1	Updating Oclusions and Visibility	122
6.1.2	Computing a Suitable View Distance	124
6.2	Evaluation	125
6.3	Discussion	125

This chapter presents SEE++, a NBV planning approach with an unstructured density representation which incorporates solutions to the challenges of proactively handling oclusions and considering scene visibility when proposing and selecting next best views. The best performing solutions to these challenges have been identified by the investigations presented in the preceding chapters as the optimisation strategy for handling oclusions and the EMR metric for selecting next best views.

SEE++ is created by integrating these solutions with SEE. The resulting approach is shown to have a significantly improved observation performance. It is capable of proposing unoccluded views from which frontier points can be more reliably observed without requiring incremental view adjustments. Next best views are selected from which greater increases in surface coverage can be attained while

moving short distances. This allows SEE++ to obtain highly complete scene observations using fewer views and shorter travel distances than other approaches while retaining the computational efficiency of an unstructured representation.

The SEE++ approach also improves on the usability of SEE by eliminating an empirical selection of the view distance. The choice of view distance is important when obtaining a scene observation as it determines the distribution of sensor measurements over scene surfaces within the viewing frustum. In the unstructured representation used by SEE and SEE++ the successful identification of frontiers in an observation depends upon the measurement density. Frontier points can be identified with the greatest reliability when the distribution of measurements obtained within the sensor viewing frustum is similar to the target measurement density. SEE++ incorporates a method for computing a suitable view distance which satisfies this constraint by considering both the target density and sensor parameters. This removes the uncertainty associated with selecting a view distance empirically and ensures that frontier points can be successfully identified in a scene observation.

The work on SEE++ discussed in this chapter was presented in Border and Gammell (2020) at the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems.

The remainder of this chapter is organised as follows. Section 6.1 details the integration of SEE with the optimisation strategy for proactively handling occlusions and the EMR view selection metric to create SEE++. Section 6.1.1 discusses two methods for selecting a subset of view proposals on which occlusion handling and updates to the frontier covisibility graph are performed. This is done to maintain a bounded computational cost when processing occlusion and visibility updates. Section 6.1.2 presents the method for computing a suitable view distance based on the target measurement density and sensor parameters. Section 6.2 presents an experimental comparison of the observation performance of SEE++ with SEE and state-of-the-art volumetric approaches. The results are discussed in Section 6.3.

The work presented in this chapter makes three key contributions:

1. SEE++, a NBV planning approach with an unstructured density representation which incorporates novel point-based techniques for proactively handling occlusions and considering scene visibility when proposing and selecting views.
2. A method for computing a suitable view distance for obtaining scene observations by considering the target measurement density and sensor parameters.
3. An experimental comparison of SEE++ with SEE and volumetric approaches showing the improvements in observation performance achieved by proposing unoccluded views and selecting next best views with high surface coverage.

6.1 SEE++

SEE++ improves upon SEE by incorporating point-based reasoning for detecting occlusions, proposing unoccluded views and considering the visibility of frontiers when selecting next best views. This allows SEE++ to retain the computational efficiency of an unstructured density representation while requiring fewer views and shorter travel distances to obtain similarly complete scene observations as SEE.

The optimisation strategy (Sec. 4.3.5) is integrated with SEE++ to proactively handle occlusions when proposing views and selecting a next best view. A frontier point is considered to be occluded from a view if there are point observations within an r -radius of the proposed sight line from the view position to the frontier point. Occlusion detection (Sec. 4.2) is performed for the τ -nearest view proposals to the current sensor position after new point measurements are obtained and processed.

Proactively handling known occlusions reduces the number of views and sensor travel distance required to observe a scene. This is the result of requiring fewer incremental view adjustments to observe frontier points as known occlusions are avoided before views are obtained. Occluded view proposals are updated to avoid known occlusions by considering the occluding points within a given radius, ψ , around each frontier point and finding the furthest sight line from any occlusion.

Scene visibility is considered when selecting a next best view by encoding the shared visibility of frontier points between views in a covisibility graph (Sec. 5.2.4). This graph connects each frontier point to the view proposals from which it can be

Algorithm 3 SEE++($\mathbf{v}_0, d, r, \rho, \tau, \psi$)

```

1:  $\mathbf{v} \leftarrow \mathbf{v}_0$                                  $\triangleright \mathbf{v}$  is the current view and  $\mathbf{v}_0$  is the initial view
2:  $\mathbf{f} \leftarrow \text{NULL}$                            $\triangleright \mathbf{f}$  is the target frontier point
3:  $\mathcal{G} \leftarrow \text{NULL}$                        $\triangleright \mathcal{G}$  is the frontier covisibility graph
4:  $P = C = F = O = \emptyset$                      $\triangleright P$  is the complete point set,  $C$  is the core set,
                                                 $F$  is the frontier set and  $O$  is the outlier set

5: while  $F \neq \emptyset$  or  $\mathbf{f} = \text{NULL}$  do
6:    $M \leftarrow \text{GET-MEASUREMENTS}(\mathbf{v})$ 
7:    $P, C, F, O \leftarrow \text{CLASSIFY-MEASUREMENTS}(M, C, F, O, r, \rho)$ 
8:   if  $F \neq \emptyset$  then
9:     if  $\mathbf{f} \in F$  then
10:       $\mathbf{v} \leftarrow \text{ADJUST-VIEW}(M, \mathbf{v}, \mathbf{f})$ 
11:       $Y \leftarrow \text{ESTIMATE-GEOMETRY}(P, F, \mathbf{v}, r)$ 
12:       $W \leftarrow \text{GET-VIEW-PROPOSALS}(Y, F, d)$ 
13:       $T \leftarrow \text{GET-NEAREST-VIEWS}(W, \mathbf{v}, \tau)$ 
14:       $Z \leftarrow \text{UPDATE-OCCLUSIONS}(T, P, \psi)$ 
15:       $\mathcal{G} \leftarrow \text{UPDATE-VIEW-GRAPH}(\mathcal{G}, Z, F, \tau)$ 
16:       $\mathbf{v}, \mathbf{f} \leftarrow \text{SELECT-NBV-EMR-METRIC}(\mathcal{G}, \mathbf{v})$ 
17: return COMPLETE

```

observed. The EMR metric (Sec. 5.3.4) is used to select a next best view from the graph that is both close to the current sensor position and has a locally maximal number of outgoing edges (i.e., visible frontiers). This restricts the travel distance of the sensor while locally maximising the coverage of partially observed scene surfaces.

The usability of SEE is improved in SEE++ by computing a suitable view distance based on the target measurement density and sensor parameters. This provides a view distance from which a sufficient measurement density can be obtained around frontiers to extend completely observed scene surfaces while ensuring that frontier points are identified at the boundaries of partially observed scene surfaces.

An overview of SEE++ is shown in Algorithm 3. As with SEE, sensor measurements are obtained and processed until there are no frontier points remaining (Lines 5–8). If the target frontier point associated with the current view is not successfully observed then the view is adjusted, as in SEE, but is not automatically chosen as the subsequent view (Lines 9–10). This allows for the consideration of alternative views with visibility of the frontier and a further adjustment of the view proposal to account for occlusions. The estimation of local surface geometry around frontier points (Line 11) and the proposal of views (Line 12) are unchanged from SEE.

A k -nearest neighbours (k -NN) search identifies a user-specified number, τ , of view proposals, T , closest to the current sensor position for which occlusion handling and updates to the covisibility graph are performed (Line 13). The motivation for limiting the number of updated view proposals and using a k -NN search to select a subset is discussed in Section 6.1.1. Occlusion detection and handling is then performed for the selected views and an updated set of view proposals, Z , is obtained (Line 14). The connectivity of each selected view in the covisibility graph, $\mathcal{G}$, is updated by evaluating the visibility of frontier points associated with the τ -nearest view proposals to the selected view and an updated graph is returned (Line 15). A next best view is then chosen from this graph using the EMR metric (Line 16).

6.1.1 Updating Occlusions and Visibility

In an ideal scenario the complete set of view proposals is evaluated when proactively handling occlusions and updating the covisibility graph. This does not impact the computation time of SEE++ in most cases as the cost per view to handle occlusions and evaluate frontier visibility is relatively small. However, this cost can increase substantially when a view is obtained of a large surface area that was previously unobserved and from which a significant number of frontier points are identified. It is typically not necessary to handle occlusions and compute covisibility for every new view proposal in this case as many of the proposed views will share similar poses and visibility of the same set of frontiers. A lower computational cost can be achieved without reducing performance by limiting updates to nearby views.

Two search-based methods were investigated for selecting a subset of view proposals based on their proximity to the current sensor position (Fig. 6.1). A radius search selects a set of view proposals within a given radius, R , of the current sensor position. This method allows the computational cost of updates to scale with the distance of the sensor from the set of proposed views. Updating fewer view proposals when the sensor is farther away is computationally efficient but reduces the fidelity of visibility information in the covisibility graph as occlusions resulting from newly obtained measurements are not accounted for when selecting a next best view.

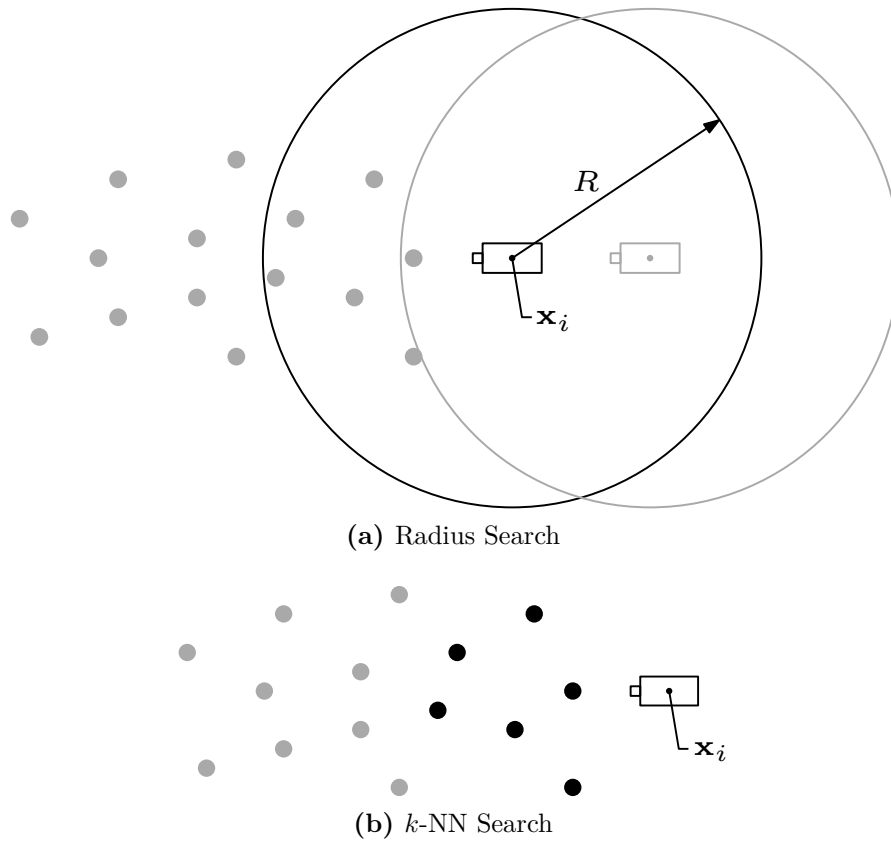


Figure 6.1: Cross-sectional illustrations of (a) selecting view proposals (grey dots) using a radius search and (b) selecting view proposals using a k -NN search. Using (a), updates are performed on all view proposals within a given radius, R , of the current sensor position, $\mathbf{x}_i$. The number of views updated depends on the radius parameter and the relative distance of view proposals from the sensor position (e.g., the grey vs black sensor), which results in varying performance and computational cost. Using (b), a fixed number of view proposals, τ , closest to the current sensor position are always selected (black dots). This method provides consistent performance and a bounded computational cost as the number of view proposals updated is independent of their distance from the current sensor position.

When the sensor is close to the set of view proposals the computational cost can increase substantially, particularly in regions with a high density of view proposals.

Using a k -NN search reduces the uncertainty in computational cost and improves the fidelity of visibility information by ensuring that a given number of view proposals, τ , close to the current sensor position are updated after every view is obtained. Consistent updates of visibility information are provided for a given number of nearby views whose visibility of frontier points may be occluded by newly obtained measurements, regardless of how far the sensor is from the set of

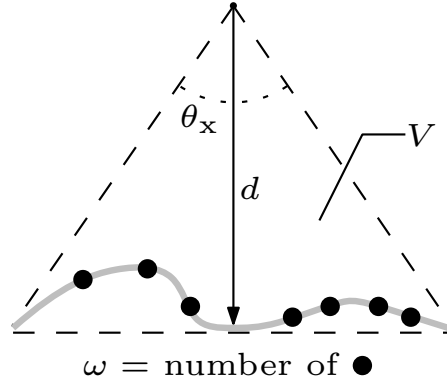


Figure 6.2: A cross-sectional illustration of the method for computing a suitable view distance, d , for observing scene surfaces (grey). The distance is computed such that the density of sensor measurements (black dots) within the volume, V , of the viewing frustum (dashed lines) is equal to the target measurement density, $\rho = \frac{\omega}{V}$.

view proposals. An upper bound on the computational cost is provided by limiting updates to a user-specified number of views. This is the method used by SEE++.

6.1.2 Computing a Suitable View Distance

The density of point measurements obtained from a given surface depends on the sensor parameters and view distance. The number of measurements obtained is defined by the sensor resolution, ω . This is typically quantified by horizontal, ω_x , and vertical, ω_y , components such that $\omega = \omega_x \omega_y$. The distribution of measurements on scene surfaces is determined by the sensor field-of-view components, θ_x and θ_y , and view distance. The field-of-view components define a viewing frustum whose surface coverage is given by the area of intersection between the frustum volume and the scene surface, which varies with distance. As the sensor moves closer to a surface the area of intersection decreases and the density of point measurements increases.

SEE and SEE++ identify frontiers in a scene observation based on changes in the density of point measurements. The ability to successfully identify frontier points from sensor measurements depends on the view distance. If the distance is too large then the point measurements will be sparsely distributed and may all be considered outlier points. When the sensor is too close to a surface the measurement density can exceed the target density and all points will be classified as core points.

A suitable view distance at which to observe a scene with a given measurement density is computed from the sensor parameters and target density (Fig. 6.2). This is the distance, d , at which the density of measurements within the viewing frustum, whose height is given by the view distance, is equal to the target density, ρ ,

$$d = \left(\frac{3\omega_x\omega_y}{4\rho \tan 0.5\theta_x \tan 0.5\theta_y} \right)^{\frac{1}{3}}. \quad (6.1)$$

The resulting view distance accounts for the existence of scene surfaces within the viewing frustum that may be closer to the sensor than the target frontier point. It ensures that the sensor is sufficiently close to surfaces to extend the scene observation while allowing new frontiers to be identified at the boundaries of partial observation.

6.2 Evaluation

The observation performance of SEE++ is compared with SEE and the previously evaluated volumetric NBV approaches by using the simulation environment discussed in Section 3.3. 100 experiments were performed with every approach on each of the standard models and the Radcliffe Camera model using the same simulated sensors (Table 3.2), view constraints (Sec. 3.3.2), algorithm parameters (Table 3.3) and performance metrics (Sec. 3.3.4) presented in Section 3.3. SEE++ uses the same view update limit and occlusion search distances presented in Section 4.4.

6.3 Discussion

SEE++ is able to achieve remarkable improvements in observation performance by considering occlusions and scene visibility when proposing and selecting next best views. The value of these considerations is demonstrated quantitatively by the experimental results (Fig. 6.3; Fig. 6.4; Table 6.1). SEE++ is shown to obtain highly complete scene observations, of a similar quality to SEE and the evaluated volumetric approaches, using significantly fewer views and shorter travel distances than the other approaches while maintaining a competitive computational time.

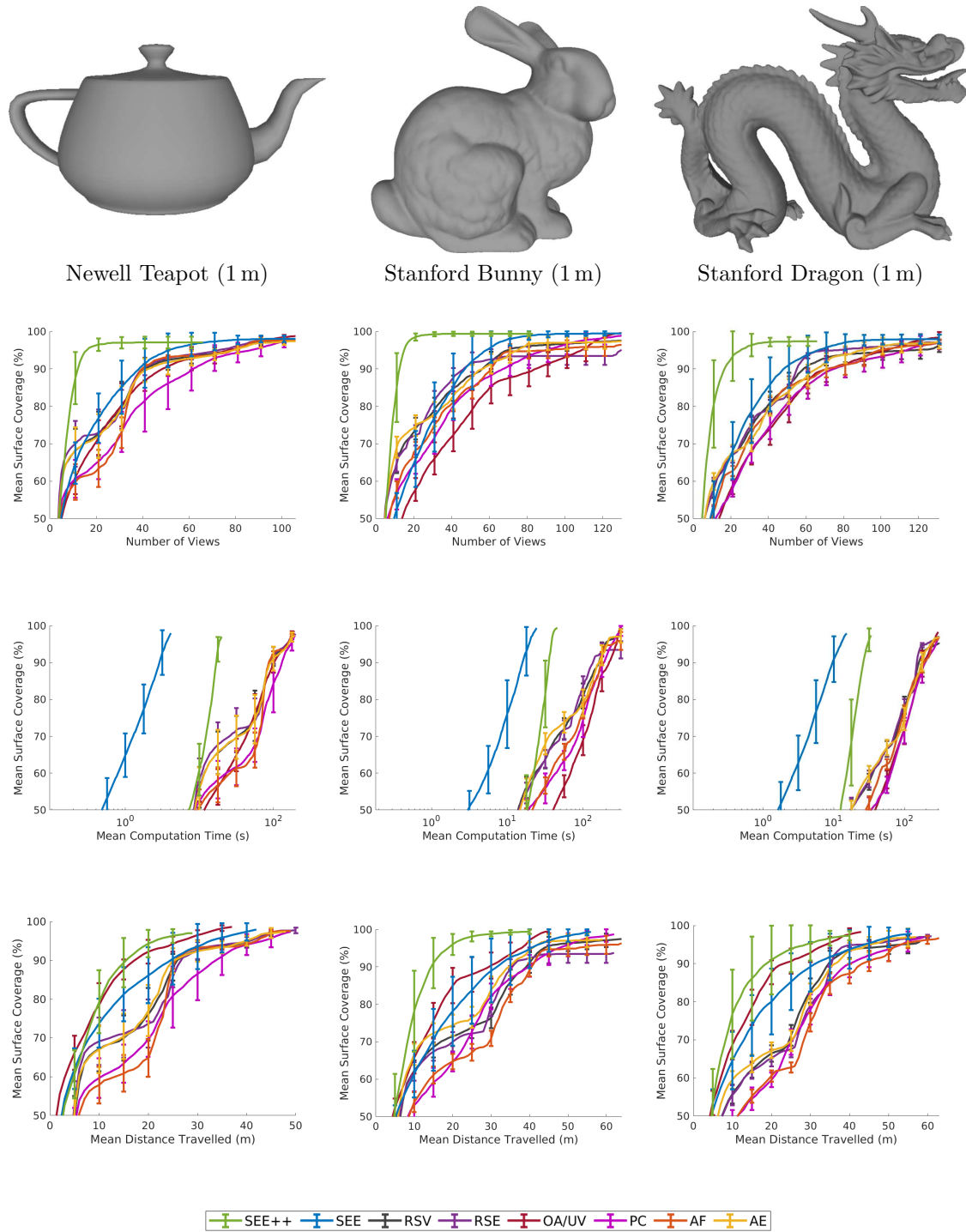


Figure 6.3: A comparison of SEE++ with SEE and the volumetric approaches. The graphs show the mean surface coverage obtained by SEE++, SEE and the volumetric approaches from 100 experiments relative to, from top to bottom, the number of views, the mean computation time and the mean travel distance.

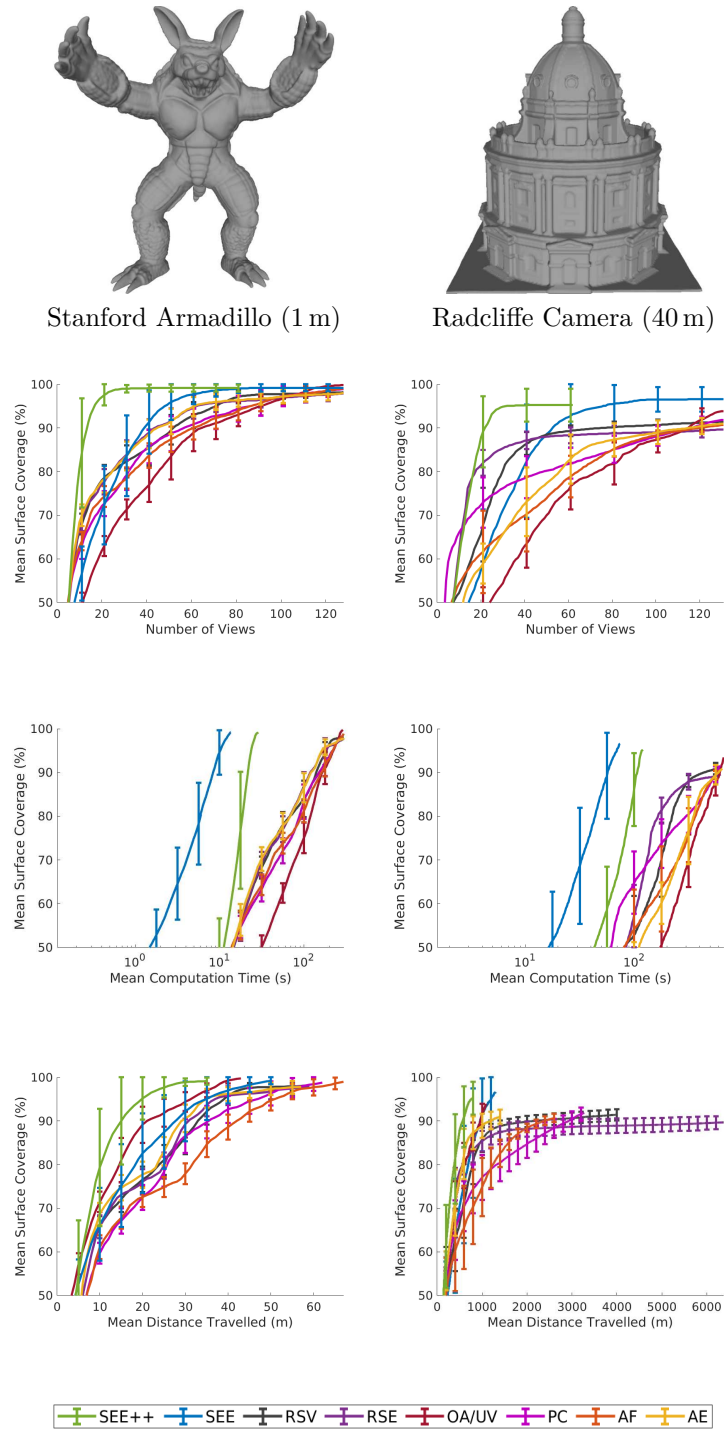


Figure 6.4: A comparison of SEE++ with SEE and the volumetric approaches. The graphs show the mean surface coverage obtained by SEE++, SEE and the volumetric approaches from 100 experiments relative to, from top to bottom, the number of views, the mean computation time and the mean travel distance.

	SEE++	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	22.3	60.0	105	105	105	105	105	105	105
Surface Coverage (%)	97.1	98.1	97.6	97.8	98.8	97.6	98.8	97.6	97.4
Computation Time (s)	20.1	4.13	196	200	194	198	194	196	196
Distance Travelled (m)	29.7	42.8	49.6	50.7	37.8	50.0	37.8	49.9	48.5

(a) Newell Teapot (Newell 1975)

	SEE++	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	31.5	75.3	129	129	129	129	129	129	129
Surface Coverage (%)	99.4	99.5	97.6	95.1	99.5	98.9	99.5	96.4	97.4
Computation Time (s)	46.0	24.7	325	324	313	320	313	326	324
Distance Travelled (m)	40.9	56.3	64.8	62.7	45.1	62.9	45.1	64.3	59.8

(b) Stanford Bunny (Turk and Levoy 1994)

	SEE++	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	35.5	78.0	130	130	130	130	130	130	130
Surface Coverage (%)	97.3	98.0	96.1	97.2	98.4	97.2	98.4	97.1	97.3
Computation Time (s)	34.0	15.4	311	311	300	306	300	306	311
Distance Travelled (m)	40.6	56.8	58.2	61.1	43.1	59.2	43.1	63.8	58.0

(c) Stanford Dragon (Curless and Levoy 1996)

	SEE++	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	30.0	63.4	127	127	127	127	127	127	127
Surface Coverage (%)	99.2	99.2	98.2	98.0	99.8	98.9	99.8	99.1	98.0
Computation Time (s)	29.1	13.7	298	301	291	298	291	298	301
Distance Travelled (m)	35.4	50.3	59.2	59.2	43.5	62.7	43.5	67.3	57.5

(d) Stanford Armadillo (Krishnamurthy and Levoy 1996)

	SEE++	SEE	RSV	RSE	UV	PC	OA	AF	AE
Number of Views	27.1	64.5	130	130	130	130	130	130	130
Surface Coverage (%)	95.3	96.6	91.4	89.7	93.9	91.9	93.9	90.7	91.0
Computation Time (s)	121	74.4	648	630	674	622	675	628	650
Distance Travelled (m)	806	1302	4008	6375	1098	3272	1098	2615	1409

(e) Radcliffe Camera (Boronczyk 2016)

Table 6.1: The mean number of views captured, the mean surface coverage obtained, the mean computation time used and the mean travel distance required to observe four one-metre standard models (Newell Teapot, Stanford Bunny, Stanford Dragon, and Stanford Armadillo) and a 40 metre model of the Radcliffe Camera, calculated from 100 experiments with SEE++, SEE and the volumetric approaches.

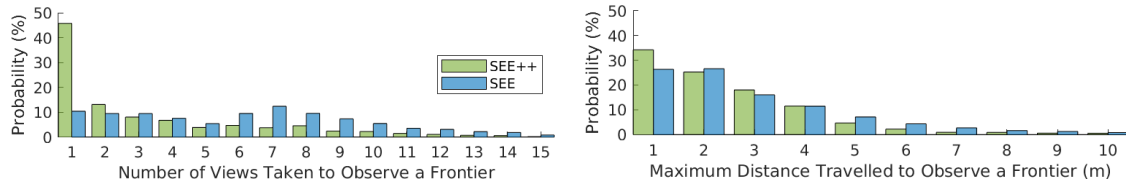


Figure 6.5: A statistical analysis of the number of views captured and maximum distance travelled in order to successfully observe frontier points using SEE and SEE++. The analysis is calculated from the standard model experimental results.

The most notable result for SEE++ is the reduction in sensor travel distance required to obtain scene observations. SEE++ is able to observe scenes while travelling less than the OA and UV volumetric approaches, which also account for occlusions. This result demonstrates the importance of considering occlusions in order to improve observation efficiency and the increased value of using a proactive strategy that can adjust views. The OA and UV approaches evaluate voxel-based occlusions when selecting a next best view but are not capable of proposing unoccluded views using this knowledge and are restricted to selecting the least occluded view from a fixed set of proposals. The fidelity of occlusion information considered is also limited by the voxel resolution. In contrast, SEE++ uses pointwise occlusion information to directly propose unoccluded views. This allows it to identify and obtain views with good frontier visibility that require lower travel distances.

The improvement in observation efficiency resulting from proactively handling occlusions with SEE++ is quantified by a statistical analysis of the number of views captured and maximum distance travelled per frontier point for SEE and SEE++ (Fig. 6.5). The analysis shows that SEE++ typically obtains successful frontier observations using fewer views and shorter travel distances than SEE. SEE++ is 4.5 times more likely to observe a frontier point with a single view than SEE. The distance travelled by SEE++ to observe a frontier point is less than 3 m in 77% of cases while SEE only observes a frontier within the same distance 69% of the time.

SEE++ directly considers the visibility of scene surfaces when selecting a next best view with the aim of obtaining the greatest improvement in surface coverage while moving short distances. The experimental results provide definitive evidence

	SEE	SEE++
Surface Coverage (%)	1.44	3.29
Distance Travelled (m)	0.75	1.23

Table 6.2: The mean obtained surface coverage and sensor travel distance per view for SEE and SEE++ calculated from the standard model experimental results.

that this goal is achieved as SEE++ is shown to obtain scene observations using considerably fewer views than SEE and the evaluated volumetric approaches. The graphs (Fig. 6.3; Fig. 6.4) visually demonstrate a rapid increase in the surface coverage attained that visibly differentiates SEE++ from the other approaches.

A statistical analysis of the mean surface coverage obtained per view for SEE and SEE++ (Table 6.2) shows that SEE++ obtains more than twice as great an increase in coverage per view as SEE. The analysis demonstrates this improvement is achieved at the cost of incurring a greater mean travel distance per view. However, the magnitude of the reduction in the number of views required is sufficient to allow scene observations to be obtained using a shorter overall travel distance than SEE.

The improvements in observation performance achieved by SEE++ are attained at the expense of an increase in computational cost. The experimental results demonstrate that in the best case, for the Radcliffe Camera, SEE++ obtains observations with a mean computation time that is 63% greater than SEE and in the worst case, for the Newell Teapot, SEE++ utilises a mean computation time that is almost five times greater than SEE. This increase in computational cost is primarily incurred by the optimisation strategy for proposing unoccluded views.

SEE++ is shown to retain a significantly greater computational efficiency than the evaluated volumetric approaches despite the increase in computation time. SEE++ can maintain a computational advantage as it is only necessary to evaluate changes in occlusions and scene visibility for views close to the current sensor position. Furthermore, the costly computation of the view optimisation strategy is only incurred for views that are found to be occluded. By contrast, the volumetric approaches perform computationally expensive raycasting to evaluate the information gain associated with every proposed view before selecting a next best view.

This chapter presents a culmination of work that leverages the strengths of an unstructured scene representation to create a NBV planning approach capable of obtaining highly complete observations with remarkable efficiency. SEE++ utilises the high-fidelity pointwise scene information encoded in the density representation of SEE to proactively handle occlusions and quantify scene visibility before obtaining a next best view. This increases the likelihood that a target frontier point will be successfully observed and that the view obtained will provide a significant improvement in the coverage of scene surfaces. SEE++ improves the usability of SEE by incorporating a method for computing a suitable view distance based on the target measurement density and sensor parameters. The experimental comparison of SEE++ with SEE and state-of-the-art volumetric NBV planning approaches demonstrates that it is capable of observing scenes using significantly fewer views and shorter travel distances than all of the other approaches while retaining a greater computational efficiency than those using a volumetric representation.

The following chapter presents a demonstration of the real world observation performance of both SEE and SEE++. Observations are obtained of several scenes with different scales and structural complexity using multiple sensing modalities. The qualitative and quantitative results presented demonstrate a successful transference of the capabilities of SEE and SEE++ from the observation of models in a simulation environment to observing real world scenes with a stereo camera and LiDAR sensor.

In summary, the work presented in this chapter makes three key contributions:

1. SEE++, a NBV planning approach with an unstructured density representation which incorporates novel point-based techniques for proactively handling occlusions and considering scene visibility when proposing and selecting views.
2. A method for computing a suitable view distance for obtaining scene observations by considering the target measurement density and sensor parameters.
3. An experimental comparison of SEE++ with SEE and volumetric approaches showing the improvements in observation performance achieved by proposing unoccluded views and selecting next best views with high surface coverage.

7

Observing the Real World

Contents

7.1	Sensor System	135
7.1.1	Intel RealSense D435	135
7.1.2	Velodyne VLP-16	137
7.1.3	Vicon System	138
7.2	Scene Observations	138
7.2.1	Single Box	139
7.2.2	Single Tower	143
7.2.3	Double Towers	146
7.2.4	Small Bookshelf	150
7.2.5	Rhinoceros Pelvis	153
7.2.6	Crocodile Skull	157
7.2.7	Summary	161
7.3	Evaluation	163
7.4	Discussion	169

This chapter presents experiments demonstrating the observation capabilities of SEE and SEE++ in the real world. Observations were obtained of several scenes with varying sizes and structural complexities using multiple different sensing modalities. Qualitative pointcloud results show that both approaches were able to obtain largely complete observations despite the presence of sensor noise. Quantitative performance metrics demonstrate the improved efficiency of SEE++ as it was able to observe all of the scenes using fewer views and shorter travel distances than SEE.

Obtaining observations of real world scenes using SEE and SEE++ introduced two key challenges that were not captured in the simulation environment: sensor noise that varies with the visual properties of the surface being observed and the need to obtain an accurate estimate of the sensor pose. The model of sensor noise used in the simulation environment is sufficient to demonstrate the ability of SEE and SEE++ to handle measurements that deviate from a true surface but does not consider failures in depth estimation that produce erroneous values or in some cases no estimate at all. When observing the real world using a given sensor it is often not possible to obtain reliable measurements from certain surfaces due to their visual properties (e.g., colouring, reflectivity or texture). This challenge was addressed in the real world experiments by applying noise filtering to sensor measurements and using parametrisations that were robust to any remaining measurement noise.

The problem of pose estimation is not considered in the simulation experiments as the sensor and model poses are explicitly specified and defined in a global coordinate frame. In the real world it is not possible to explicitly define the sensor pose within a virtual coordinate frame. Instead it is necessary to estimate the sensor pose relative to a known reference frame. The relative pose of a sensor can be estimated from its egomotion (i.e., its movement relative to the scene structure observed with sensor measurements) or using an external tracking system. Using an egomotion approach to estimate the sensor pose allows scenes to be observed without an external system but requires the use of a relative coordinate frame.

Egomotion estimation tracks incremental pose changes between sensor measurements and represents the current sensor pose relative to the initial pose of the sensor by accumulating the incremental transformations. This technique is typically prone to increasing drift in the sensor pose due to the accumulation of errors from each incremental estimate and requires a bounding box for the scene to be specified relative to the initial sensor pose. Developing better techniques to estimate relative egomotion is an active area of research. To isolate the performance of SEE and SEE++ from the ongoing research into relative pose estimation the sensor pose was estimated using an external tracking system that defined a known coordinate frame.

The scenes were observed independently using either a stereo camera or a LiDAR sensor. This demonstrates the ability of SEE and SEE++ to generalise between sensing modalities with different capabilities. Stereo cameras can typically obtain accurate measurements of surfaces with unique visual features at short distances but are unable to obtain reliable measurements, if a measurement can be obtained at all, from surfaces with uniform or repetitive features and surfaces farther from the sensor.

LiDAR sensors can obtain accurate measurements at longer ranges than stereo cameras as their depth estimation does not depend on the identification of unique visual features. The noise associated with LiDAR measurements typically has a lower variability than stereo cameras when observing most surfaces, except when observing surfaces with high reflectivity. The sensor resolution is lower than that of stereo cameras and measurements are obtained along scan lines, resulting in a non-uniform distribution of measurements within the sensor field-of-view.

The stereo camera and LiDAR sensor were attached to a handheld *sensor wand* together with a set of markers whose position and relative configuration defined a coordinate frame for the wand which could be tracked by an external Vicon system (Sec. 7.1). The Vicon defined a coordinate frame with a known origin in the real world that was used to specify bounding boxes for the scene observations.

The algorithms are demonstrated on six different scenes (Sec. 7.2). The visual properties, surface complexity and size of each scene is varied, which presented unique challenges when obtaining observations using the different sensors. Some scenes contain surfaces with uniform or repetitive features which proved difficult to observe with the stereo camera. The reflectivity of surfaces in some scenes reduced the number of reliable measurements obtained with the LiDAR sensor. The varying complexity of the surface geometry provides a useful distinction between the capabilities of SEE and SEE++ as scenes with self-occluding surfaces were typically observed much more efficiently using SEE++. The difference in scene sizes demonstrates the considerations necessary to obtain highly complete observations at any scale and the ability of SEE and SEE++ to generalise with scene size.

The qualitative pointcloud observations obtained for each scene with SEE and SEE++, separately using the stereo camera and LiDAR sensor, are presented in the corresponding subsections of Section 7.2 and discussed in Section 7.2.7. A quantitative evaluation of the observation performance for SEE and SEE++ is presented in Section 7.3. The overall outcomes of the experiments are discussed in Section 7.4.

The work presented in this chapter makes three key contributions:

1. Real world experiments demonstrating the observation capabilities of SEE and SEE++ on several real world scenes with varying sizes and structural complexities independently using either a stereo camera or LiDAR sensor.
2. Qualitative pointcloud results showing that both SEE and SEE++ were able to obtain largely complete observations despite the presence of sensor noise.
3. Quantitative metrics of the observation performance for SEE and SEE++ demonstrating that the efficiency improvements of SEE++ enabled it to observe all of the scenes using fewer views and with less travelling than SEE.

7.1 Sensor System

The scenes were observed using a handheld sensor ‘wand’ (Fig. 7.1). Measurements were obtained using either the Intel RealSense D435 or the Velodyne VLP-16 attached to the wand. A Vicon system was used to track the pose of this sensor wand relative to a known coordinate frame defined in the scene workspace. Observations were obtained by moving the sensor wand to the poses of next best views selected by SEE or SEE++ and capturing sensor measurements. A 3D viewer was used to visualise the current pointcloud observation, the real-time pose of the sensor wand and the chosen next best view pose. New measurements were obtained when the tracked sensor pose was within a thresholded offset of the next best view pose.

7.1.1 Intel RealSense D435

The Intel RealSense D435 is a stereo camera that obtains depth measurements using a stereo pair of infrared sensors. Unique features are identified in the images obtained from each sensor and matching is performed between the two sets of features. Feature

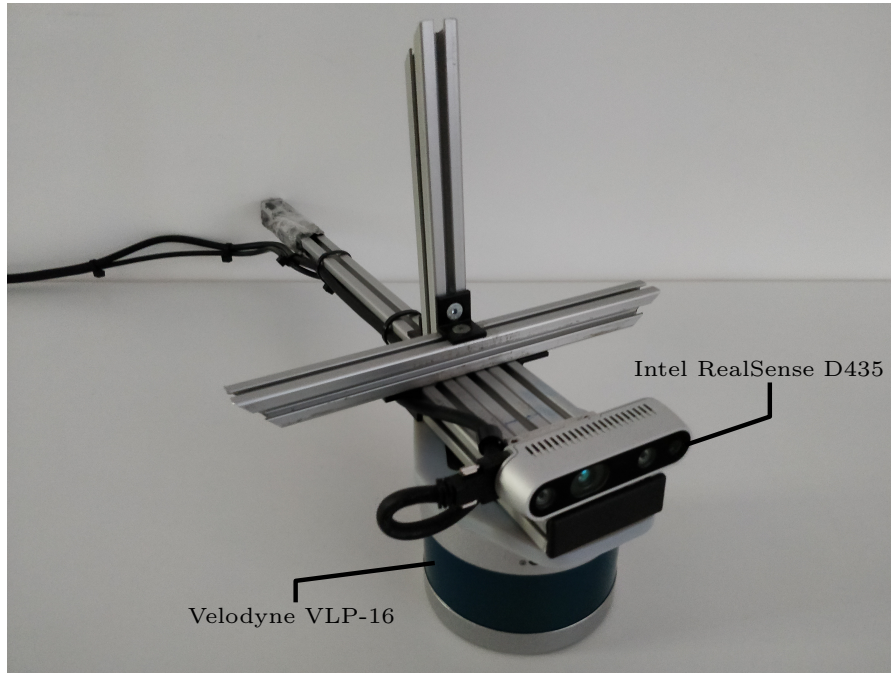


Figure 7.1: The sensor system used to observe the real world scenes. Observations were obtained independently using either an Intel RealSense D435 or a Velodyne VLP-16. The sensor poses were tracked using a Vicon motion capture system. Pose estimation is performed by identifying the position and relative configuration of reflective vicon markers in the images of 10 infrared cameras. The sensors and vicon markers (not shown) were affixed to a metal ‘sensor wand’ in a known configuration.

matches are used in conjunction with the known separation between the sensors to compute a depth measurement using triangulation. An infrared projector is used to improve the estimated depth by projecting unique features onto scene surfaces, but this is only effective at short distances. Depth measurements are combined with colour pixels obtained from an RGB sensor to produce a coloured pointcloud.

Spatial and temporal filtering are applied to depth measurements obtained using the RealSense in order to reduce sensor noise and eliminate erroneous points. Spatial filtering is performed using a decimation filter provided with the RealSense SDK. This obtains the median value for 2x2 pixel regions in the depth image to smooth over erroneous measurements and effectively halves the sensor resolution.

The RealSense data were also filtered temporally to remove inconsistent measurements between observations. The temporal filter evaluates the correspondence between pointclouds using a similar method to the Iterative Closest Point (ICP)

algorithm (Yang and Medioni 1992; Besl and McKay 1992). A one-to-one matching is computed between points in subsequently obtained pointclouds, P_t and P_{t+1} . A matching point in the P_{t+1} pointcloud is found for a point in the P_t pointcloud by evaluating unmatched points within a given filter distance, f . Points in the P_{t+1} pointcloud that are not matched with a point in P_t are removed. Temporal filtering is applied iteratively for a given number of subsequent sensor observations, n , in order to obtain a pointcloud containing measurements consistent between all observations.

Sensor measurements were obtained using the RealSense with a resolution of $w_x = 848$ px, $w_y = 480$ px and a field-of-view of $\theta_x = 69.4^\circ$, $\theta_y = 42.5^\circ$. The spatial filtering applied to the depth measurements reduced the effective resolution of the pointclouds obtained by SEE and SEE++ to $w_x = 424$ px, $w_y = 240$ px. A filter distance of $f = 0.005$ m was used for the temporal filtering of $n = 30$ subsequent pointclouds. This represents 1 s of observations as the sensor framerate used was 30 Hz.

7.1.2 Velodyne VLP-16

The Velodyne VLP-16 is LiDAR sensor that obtains depth measurements based on the time-of-flight of an infrared pulse emitted from the sensor and reflected back from a scene surface. The time delay between emitting a pulse and receiving a corresponding reflection is used to compute the distance to a scene surface and the reflection intensity can be used to identify surface properties. The sensor contains a rotating mechanism that allows it to obtain measurements with a 360° horizontal field-of-view. This rotates at 10 Hz with a horizontal angular resolution of $\approx 0.25^\circ$ and obtains 16 measurements in a 30° vertical field-of-view from each position.

Measurements obtained using the Velodyne were filtered by restricting the sensor field-of-view and applying temporal filtering. The horizontal field-of-view is centred on the x -axis in the sensor coordinate frame, which points directly forwards on the handheld sensor wand (Fig. 7.1). This field-of-view was restricted to 90° so that the measurements obtained from rotational positions less than -45° or greater than $+45^\circ$ were not included in the sets of measurements obtained by SEE and SEE++. Temporal filtering was performed using the approach discussed in Section 7.1.1.

The resulting field-of-view for Velodyne measurements obtained by SEE and SEE++ was $\theta_x = 90^\circ$, $\theta_y = 30^\circ$. The effective resolution of the sensor within this field-of-view was computed from its specifications as $w_x = 586$ px, $w_y = 16$ px. A filter distance of $f = 0.01$ m was used for the temporal filtering of $n = 10$ subsequent pointclouds. This represents 1 s of observations from a sensor rotating at 10 Hz.

7.1.3 **Vicon System**

A Vicon motion capture system was used to obtain pose estimates for the sensor wand by identifying the position and relative configuration of reflective markers affixed to the wand. Ten cameras placed around the Vicon workspace were used to identify the marker positions from infrared images based on the intensity of reflected light. A coordinate frame is defined for the sensor wand based on the unique configuration of attached markers. The relative transformations between this frame and the sensor coordinate frames were obtained using a manual calibration.

7.2 **Scene Observations**

The real world observation performance of SEE and SEE++ is demonstrated on six different scenes. Observations of each scene were obtained with each approach independently using either the Intel RealSense D435 or the Velodyne VLP-16. The structural and visual properties of each scene present different challenges when obtaining observations with the two sensing modalities.

The single box (Sec. 7.2.1), single tower (Sec. 7.2.2) and double tower (Sec. 7.2.3) scenes contain foam boxes in various configurations. The small bookshelf scene (Sec. 7.2.4) consists of a bookshelf with two shelves full of books. The rhinoceros pelvis (Sec. 7.2.5) and crocodile skull (Sec. 7.2.6) scenes contain specimens loaned from the Oxford University Museum of Natural History.

The observations of each scene obtained using SEE and SEE++ are evaluated by considering qualitative pointcloud results and quantitative metrics of the observation performance. Figures are presented in the following subsections showing photographs of each scene and representative views of the pointcloud results. A complete set

photographs and views are included in Appendix A. A summative assessment of the qualitative pointcloud results is presented in Section 7.2.7.

7.2.1 Single Box

The single box scene contains a foam box with dimensions 0.6x0.6x0.6m. The box sides consist of coloured foam panels marked with unique numbers in reflective black duct tape. The bounding box of the scene is sufficiently large to encompass a region of the floor area around the box (Fig. 7.2; Fig. A.1). The surface geometry of the scene is simple and contains no major occlusions but certain visual properties, such as the reflectivity of the duct tape and the repetitive texture of the box edges, often prevented reliable measurements of these surface from being captured.

7.2.1.1 RealSense

The RealSense observations were obtained using a target measurement density of 500000 points per m^3 and a resolution of 0.05 m. The qualitative pointcloud results (Fig. 7.3; Fig. A.2 and Fig. A.3) show that both SEE and SEE++ obtained largely complete observations except for in the numerical markings and along the box edges as accurate measurements of these features could not be obtained from many views. The numerical markings frequently returned noisy measurements due to the reflective black surface of the duct tape. Due to the uncertainty of these points they were often removed by the temporal filtering.

When the box edges were observed from views orthogonal to the box sides the stepwise difference in depth between the boundary of the box surface and the scene background produced significant measurement noise. Accurate measurements of the edges could often only be obtained from views with visibility of both sides of the surface discontinuity. The results show that SEE was more likely to obtain accurate measurements of the box edges than SEE++. This can be attributed to the greater number of adjusted views captured by SEE as these were less likely to be orientated orthogonally to the box sides and provided better visibility of the box edges.

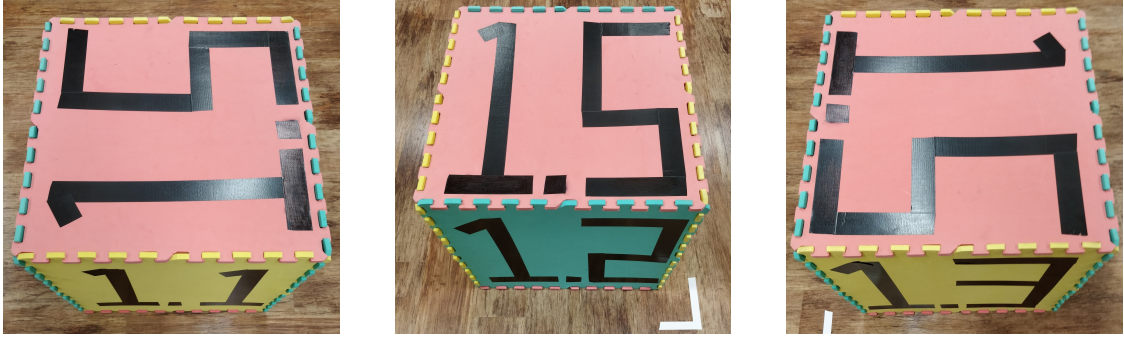


Figure 7.2: Photographs of the single box scene. It contains a 0.6x0.6x0.6m foam box with a unique numerical marking on each side in reflective black duct tape.

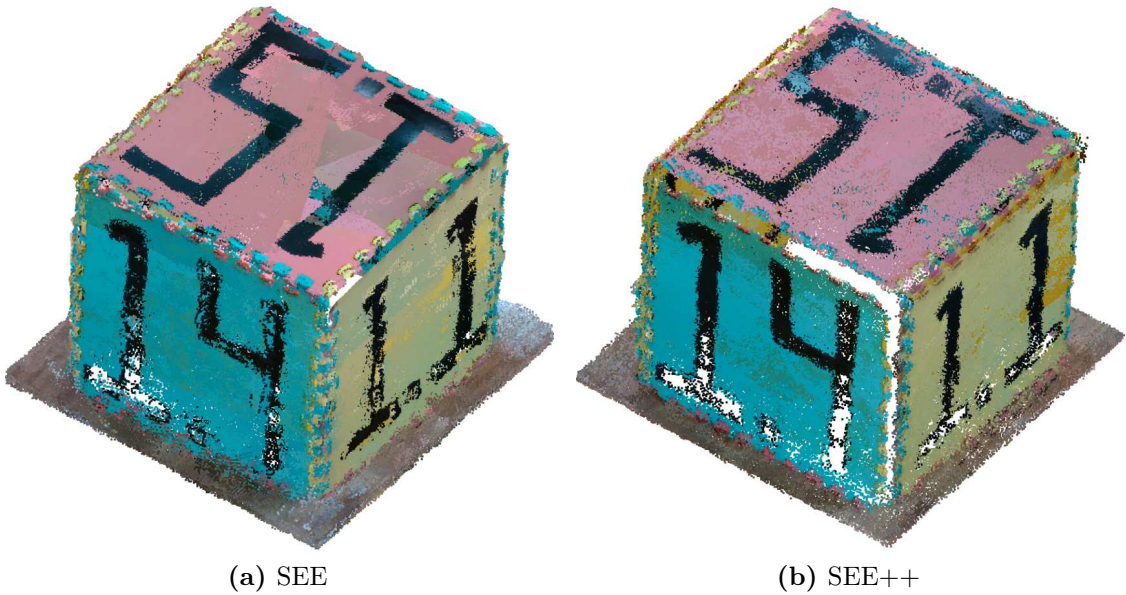


Figure 7.3: The RGB coloured pointcloud results obtained from observations of the single box scene using the Intel RealSense D435 with SEE and SEE++.

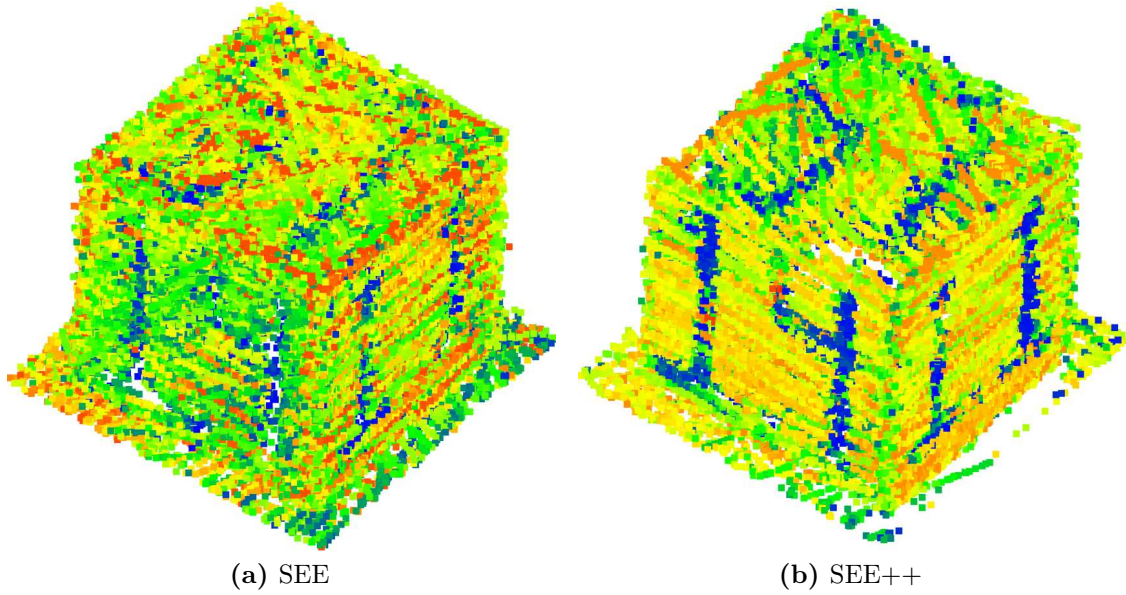


Figure 7.4: The LiDAR intensity coloured pointcloud results obtained from observations of the single box using the Velodyne VLP-16 with SEE and SEE++.

Sensors	RealSense		Velodyne	
Resolution (m), r	0.05		0.1	
Target Density, ρ	500000		20000	
Algorithms	SEE	SEE++	SEE	SEE++
Number of Views	81	68	160	96
Travel Distance (m)	33	25	86	33
Computation Time (s)	118	144	26	65

Table 7.1: Top: the resolution radius, r , and target measurement density, ρ , used for observations of the single box scene with the Intel RealSense D435 and the Velodyne VLP-16. Bottom: the number of views, travel distance and computation time required to obtain scene observations with SEE and SEE+ using each sensor.

The quantitative performance metrics (Table 7.1) demonstrate that SEE++ obtained an observation using 16% fewer views and a 24% shorter travel distance than SEE while requiring a 22% increase in computation time. As this scene contains no major occlusions the improved performance of SEE++ can primarily be attributed to the consideration of scene visibility when selecting next best views. The increased computational cost is incurred by maintaining the covisibility graph.

7.2.1.2 Velodyne

The Velodyne observations were obtained using a target measurement density of 20000 points per m^3 and a resolution of 0.1 m. The qualitative pointcloud results (Fig. 7.4; Fig. A.4 and Fig. A.5) show that both SEE and SEE++ obtained largely complete observations except for on the numerical markings and in the case of SEE++ the floor area on one side of the box. It was often not possible for the Velodyne to capture reliable measurements from the black duct tape due to its reflectivity. Accurate measurements could often only be obtained from views orthogonal to the box sides as this relative orientation reduced the likelihood of adverse reflections. The distinct visibility of the numerical markings in the SEE++ pointcloud result demonstrate that it was able to obtain more accurate measurements from the duct tape than SEE by obtaining a greater proportion of orthogonal views. The absence of measurements in the SEE++ pointcloud from the floor area on one side of the box indicates that it was able to obtain a sufficient measurement density without observing that region. This is likely possible due to the relatively small width of the region in comparison to the resolution radius.

The quantitative performance metrics (Table 7.1) demonstrate that SEE++ obtained an observation using 40% fewer views and a 62% shorter travel distance than SEE. The computation time used by SEE++ was 2.5 times greater than required by SEE. This shows that SEE++ was able to achieve a significant improvement in observation performance by considering scene visibility at the expense of an increased computational cost. The relative increase in computation time is greater than was incurred when observing the scene with the RealSense. This is because

the sparse measurements obtained by the Velodyne were processed in a shorter computation time but the cost of updating the visibility graph was similar to the RealSense observation as approximately the same number of views were represented.

7.2.2 Single Tower

The single tower scene contains two foam boxes in a stacked configuration. The top box is rotated by 45° around the z -axis relative to bottom box. The scene bounding box is aligned with the rotation of the top box such that a square region of floor area, offset from the bottom box by 45° , is encompassed (Fig. 7.5; Fig. A.6). The rotational offset between the boxes creates small regions of self-occlusion from the lower corners of the top box but the surface geometry remains fairly simple.

7.2.2.1 RealSense

The RealSense observations were obtained using a target measurement density of 500000 points per m^3 and a resolution of 0.05 m. The qualitative pointcloud results (Fig. 7.6; Fig. A.7 and Fig. A.8) for both SEE and SEE++ are largely complete with the exceptions of gaps that can be observed along the box edges and in the numerical markings. The SEE++ pointcloud is shown to contain more significant gaps along the edges than are present in the SEE pointcloud. As discussed for the single box scene this can be attributed to the less frequent use of view adjustment by SEE++, which reduces the number of views with sufficient visibility of the box edges to obtain reliable measurements. The lower level of measurement noise in the SEE++ pointcloud is likely due to the use of more orthogonal views. The numerical markings are also marginally more complete than in the SEE pointcloud.

The quantitative performance metrics (Table 7.2) demonstrate that SEE++ obtained an observation using 34% fewer views and a 43% shorter travel distance than SEE at the expense of incurring a 15% increase in computation time. While the scene contains some small self-occlusions, the majority of this performance improvement and increased computational cost can be attributed to the consideration of scene visibility when selecting next best views with SEE++.



Figure 7.5: Photographs of the single tower scene. It contains two foam boxes in a stacked configuration with a 45° rotational offset between the boxes.

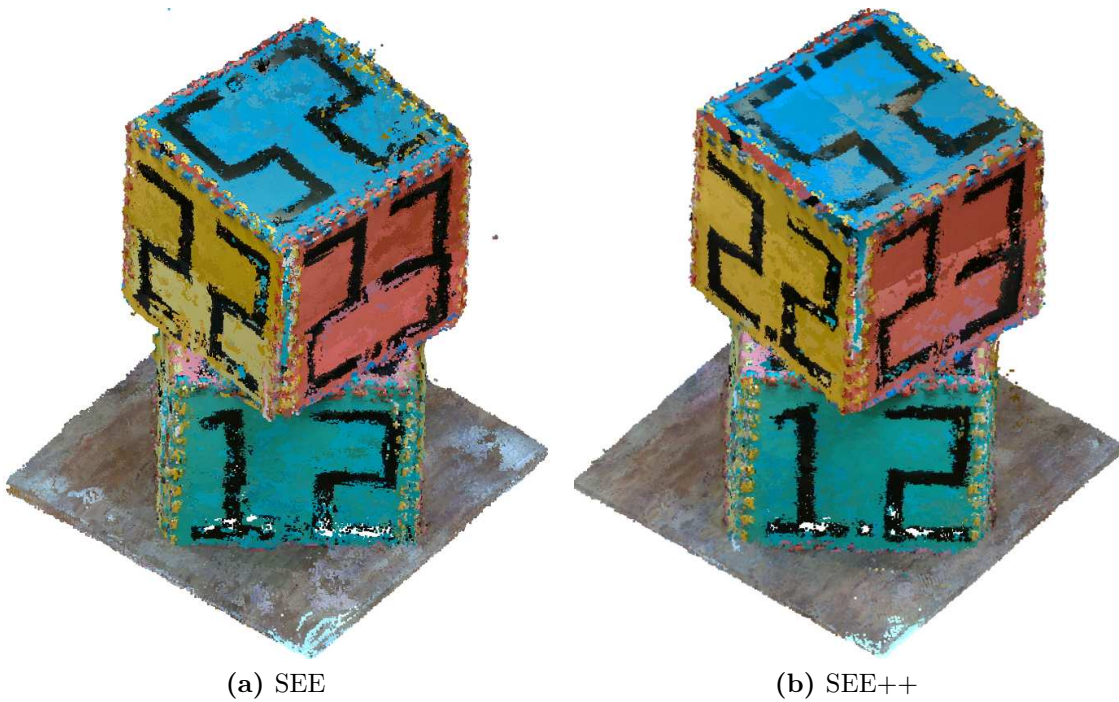


Figure 7.6: The RGB coloured pointcloud results obtained from observations of the single tower scene using the Intel RealSense D435 with SEE and SEE++.

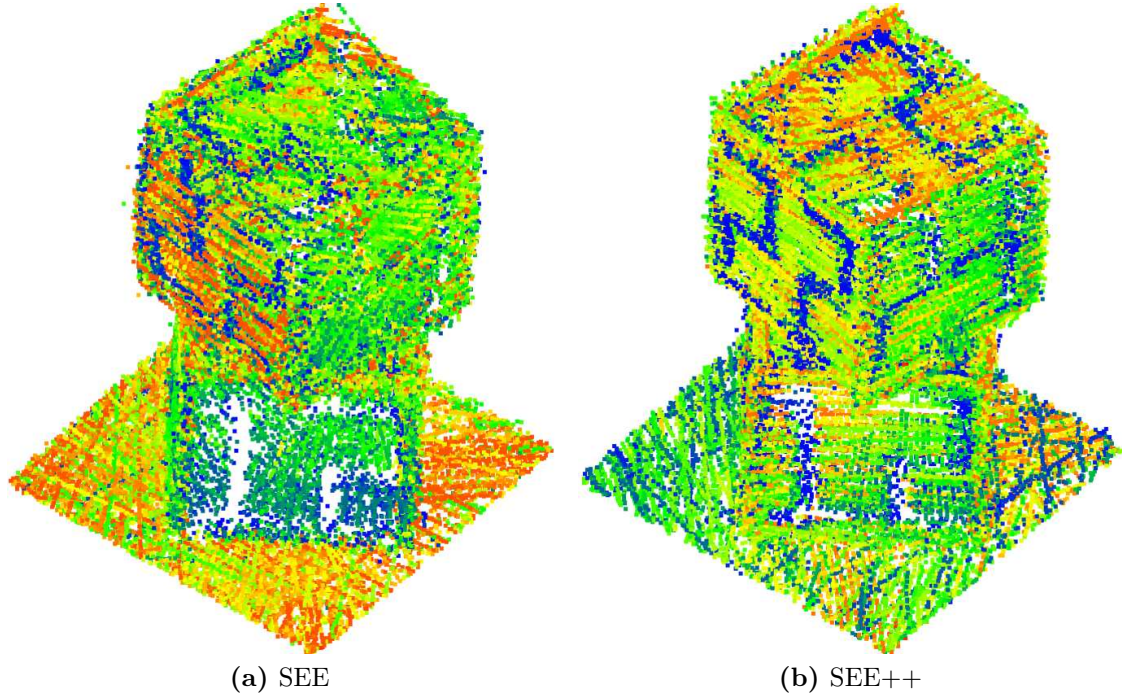


Figure 7.7: The LiDAR intensity coloured pointcloud results obtained from observations of the single tower using the Velodyne VLP-16 with SEE and SEE++.

Sensors	RealSense		Velodyne	
Resolution (m), r	0.05		0.1	
Target Density, ρ	500000		20000	
Algorithms	SEE	SEE++	SEE	SEE++
Number of Views	207	136	155	104
Travel Distance (m)	89	51	54	47
Computation Time (s)	261	301	32	58

Table 7.2: Top: the resolution radius, r , and target measurement density, ρ , used for observations of the single tower scene with the Intel RealSense D435 and the Velodyne VLP-16. Bottom: the number of views, travel distance and computation time required to obtain scene observations with SEE and SEE+ using each sensor.

7.2.2.2 Velodyne

The Velodyne observations were obtained using a target measurement density of 20000 points per m^3 and a resolution of 0.1 m. The qualitative pointcloud results (Fig. 7.7; Fig. A.9 and Fig. A.10) show that both SEE and SEE++ were able to obtain highly complete observations of the box structure and floor region without significant measurement noise. However, the measurements obtained from the reflective numerical markings were often noisy and therefore removed by temporal filtering. The greater completeness of the SEE++ pointcloud demonstrates that it was more successful at obtaining accurate measurements of the numerical markings as it utilised fewer view adjustments and obtained more orthogonal views.

The quantitative performance metrics (Table 7.2) demonstrate that SEE++ obtained an observation using 33% fewer views than SEE and a 13% shorter travel distance than SEE. The computation time used by SEE++ was 1.8 times greater than required by SEE. These results demonstrate a lower reduction in the travel distance for SEE++ over SEE than for the single box scene. This is because SEE was able to observe this scene while travelling less than for the single box scene but SEE++ travelled farther to observe this scene than the single box scene.

7.2.3 Double Towers

The double towers scene contains two sets of stacked boxes separated by a distance of less than 1 m (Fig. 7.8; Fig. A.11). The stacked boxes have offset rotations as in the single tower scene (Sec. 7.2.2). The separation distance between the two box towers produces a scene with several occluding surfaces. The view distance used to obtain observations with either sensor is greater than this separation distance and as such is not possible to observe the inward facing side of either top box from a view orthogonal to the surface.

7.2.3.1 RealSense

The RealSense observations were obtained using a target measurement density of 100000 points per m^3 and a resolution of 0.1 m. The desired measurement density



Figure 7.8: A photograph of the double towers scene. It contains two stacks of rotationally offset foam boxes separated by a distance of less than 1 m.



Figure 7.9: The RGB coloured pointcloud results obtained from observations of the double towers scene using the Intel RealSense D435 with SEE and SEE++.

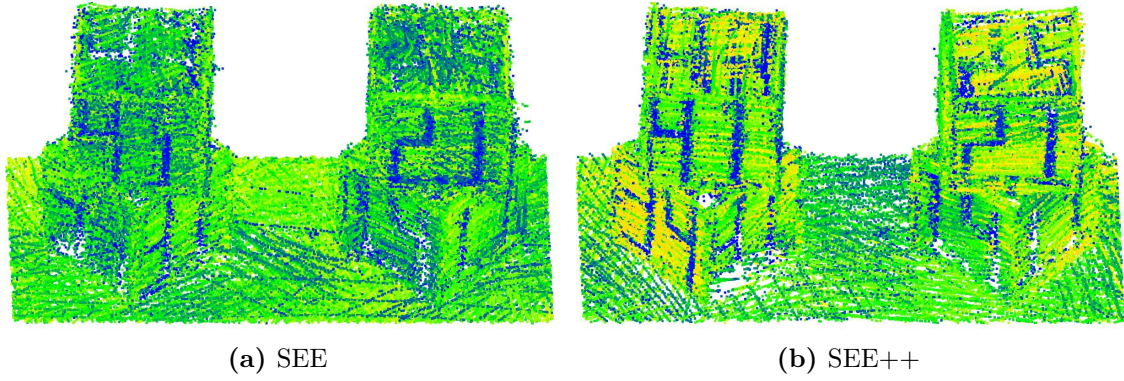


Figure 7.10: The LiDAR intensity coloured pointcloud results obtained from observations of the double towers using the Velodyne VLP-16 with SEE/SEE++.

Sensors	RealSense		Velodyne	
Resolution (m), r	0.1		0.1	
Target Density, ρ	100000		20000	
Algorithms	SEE	SEE++	SEE	SEE++
Number of Views	125	93	213	148
Travel Distance (m)	67	37	110	54
Computation Time (s)	415	562	60	163

Table 7.3: Top: the resolution radius, r , and target measurement density, ρ , used for observations of the double towers scene with the Intel RealSense D435 and the Velodyne VLP-16. Bottom: the number of views, travel distance and computation time required to obtain scene observations with SEE and SEE+ using each sensor.

was decreased and a larger resolution was used in order to handle an increase in measurement noise. A greater number of noisy measurements were obtained as the larger scene scale increased the distance between scene surfaces and the sensor. The large floor area visible in the scene also produced an increase in measurement noise.

The qualitative pointcloud results (Fig. 7.9; Fig. A.12 and Fig. A.13) obtained by SEE and SEE++ show this increase in measurement noise and the larger gaps in the observations resulting from the decreased measurement density and increased resolution radius. The level of noise is marginally greater in the SEE++ pointcloud. This is likely the result of capturing measurements from views proposed by the optimisation strategy that provided visibility of the inward facing box sides but increased the measurement noise due to the angle at which the surfaces were observed.

The quantitative performance metrics (Table 7.3) demonstrate that SEE++ obtained an observation using 26% fewer views and a 45% shorter travel distance than SEE while requiring a 35% increase in computation time. This illustrates the significant improvement in observation performance that can be achieved when observing scenes containing major occlusions by using proactive occlusion handling and considering scene visibility. The cost of this improvement is an increase in computation time that in this case is primarily incurred by using the optimisation strategy to propose unoccluded views.

7.2.3.2 Velodyne

The Velodyne observations were obtained using a target measurement density of 20000 points per m^3 and a resolution of 0.1 m. The Velodyne measurement noise remained consistent with the smaller scenes as it does not vary significantly with distance and therefore the same parameters were used. The qualitative pointcloud results (Fig. 7.10; Fig. A.14 and Fig. A.15) show that both SEE and SEE++ obtained largely complete and accurate observations. As with the smaller scenes, reliable measurements could often not be obtained from the numerical markings. SEE++ captured sparser measurements of some scene regions than SEE. This

indicates that reliable measurements of these surfaces could not be obtained from the views captured by SEE++. This is likely the result of obtaining views proposed by the optimisation strategy that enabled visibility but observed the surfaces from angles that increased the measurement noise.

The quantitative performance metrics (Table 7.3) demonstrate that SEE++ obtained an observation using 31% fewer views and a 51% shorter travel distance than SEE. SEE++ required a 2.7 times greater computation time than SEE to observe the scene. This illustrates that a significant improvement in observation performance was achieved by proactively handling occlusions and considering scene visibility when proposing and selecting next best views. The greater efficiency of observations was attained at the expense of an increased computational cost.

7.2.4 Small Bookshelf

The small bookshelf scene contains a bookshelf with two shelves full of books (Fig. 7.11; Fig. A.16). The surface geometry of the bookshelf itself is relatively simple, but the gap between each row of books and the top of their shelf produced a configuration of occluded surfaces that were challenging to observe using either sensor. Measurements could be obtained from the top of the books and the back of the shelves but this was only possible from certain view orientations.

7.2.4.1 RealSense

The RealSense observations were obtained using a target measurement density of 250000 points per m^3 and a resolution of 0.05 m. The desired measurement density was decreased to account for the sparsity of measurements obtainable from the gap between the books and the top of the shelves. The qualitative pointcloud results (Fig. 7.12; Fig. A.17 and Fig. A.18) show that both SEE and SEE++ were able to obtain largely complete observations despite the presence of significant sensor noise, which produced multiple offset surface measurements when observing the floor and outer sides of the bookshelf.



Figure 7.11: Photographs of the small bookshelf scene. It consists of two shelves full of books. The photographs show two angled side views of the bookshelf.



Figure 7.12: The RGB coloured pointcloud results obtained from observations of the small bookshelf scene using the Intel RealSense D435 with SEE and SEE++.

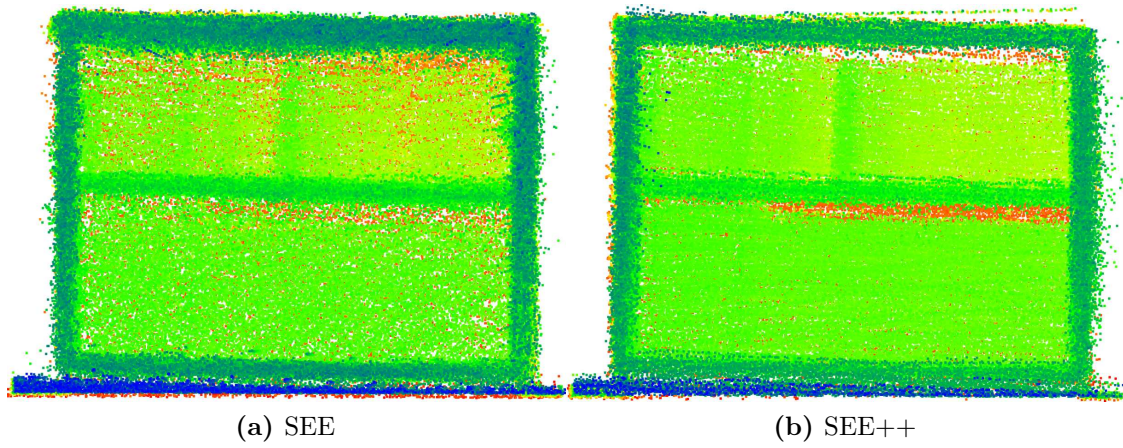


Figure 7.13: The y -coordinate coloured pointcloud results obtained from observations of the small bookshelf using the Velodyne VLP-16 with SEE and SEE++.

Sensors	RealSense		Velodyne	
Resolution (m), r	0.05		0.1	
Target Density, ρ	250000		20000	
Algorithms	SEE	SEE++	SEE	SEE++
Number of Views	177	76	178	138
Travel Distance (m)	120	28	105	80
Computation Time (s)	242	287	30	59

Table 7.4: Top: the resolution radius, r , and target measurement density, ρ , used for observations of the small bookshelf scene with the Intel RealSense D435 and the Velodyne VLP-16. Bottom: the number of views, travel distance and computation time required to obtain scene observations with SEE and SEE+ using each sensor.

The quantitative performance metrics (Table 7.4) demonstrate that SEE++ observed the scene using 57% fewer views and a 77% shorter travel distance than SEE while requiring a 19% increase in computation time. This illustrates the importance of utilising proactive occlusion handling when observing scenes that contain major self-occlusions. It shows that by doing so the improvement in observation performance achieved, in terms of a reduction in the number of views and travel distance required, can be significant enough for a scene to be observed without incurring a correspondingly large increase in computational cost.

7.2.4.2 Velodyne

The Velodyne observations were obtained using a target measurement density of 20000 points per m^3 and a resolution of 0.1 m. The qualitative pointcloud results (Fig. 7.13; Fig. A.19 and Fig. A.20) demonstrate that SEE++ was able to obtain more accurate measurements from the gaps above the books by considering the occluding surfaces when proposing views. This is shown by a more clearly defined difference in colour for the gaps between the rows of books and the backs of the shelves.

The quantitative performance metrics (Table 7.4) demonstrate that SEE++ observed the scene using 22% fewer views and a 24% shorter travel distance than SEE. The computation time used by SEE++ was two times greater than required by SEE. This is a less significant improvement in observation performance than was achieved for the RealSense observations. It can be attributed to the increased sparsity of measurements obtained using the Velodyne as this meant it was necessary to capture more views of the bookshelf before its structure could be observed with sufficient fidelity to identify the configuration of self-occlusions between the books and shelves.

7.2.5 Rhinoceros Pelvis

The rhinoceros pelvis scene (Fig. 7.14; Fig. A.21) contains the pelvis of a Javan Rhinoceros (*rhinoceros sondaicus*) specimen loaned from the Oxford University Museum of Natural History (OUMNH 19164). The detail of visual features and

surface texture on the pelvis in this scene is significantly greater than for the foam box and bookshelf scenes. The relatively complex geometry of the pelvis produces many self-occlusions but also contains large surface areas with continuous geometry. This made some surfaces difficult to observe while also enabling great increases in scene coverage from certain views. The absence of reflective surfaces and the presence of unique visual features reduced the measurement noise for both sensors.

7.2.5.1 RealSense

The RealSense observations were obtained using a target measurement density of 500000 points per m^3 and a resolution of 0.05 m. The qualitative pointcloud results (Fig. 7.15; Fig. A.22 and Fig. A.23) show that both SEE and SEE++ were able to obtain highly complete observations. SEE obtained sparser measurements than SEE++ from some scene surfaces (e.g., Fig. 7.15c). This is likely because SEE++ was able to capture more reliable measurements of these surfaces from views with a more orthogonal orientation to the local surface geometry.

The quantitative performance metrics (Table 7.5) demonstrate that SEE++ observed the scene using 15% fewer views and a 17% shorter travel distance than SEE. SEE++ incurred a computation time 2.7 times greater than SEE in order to attain this improvement in observation performance. This is a greater proportional increase in computational cost than occurred for the RealSense observations of the larger scenes. There is a greater increase in computation time as SEE obtains an observation of the scene using fewer views than were required for the larger scale scenes while SEE++ does not achieve an equivalent reduction in the number of views obtained and incurs an increased computational cost due to a more frequent use of view optimisation. This can be attributed to the complex surface geometry.

7.2.5.2 Velodyne

The Velodyne observations were obtained using a target measurement density of 20000 points per m^3 and a resolution of 0.05 m. It was possible to use a smaller resolution radius for this scene due to the reduction in measurement noise. The qualitative pointcloud results (Fig. 7.16; Fig. A.24 and Fig. A.25) show that SEE



Figure 7.14: Photographs of the rhinoceros pelvis scene. It contains the pelvis of a Javan rhinoceros (*rhinoceros sondaicus*) specimen that was loaned from the Oxford University Museum of Natural History (OUMNH 19164).



(a) SEE (front)



(b) SEE++ (front)



(c) SEE (back)



(d) SEE++ (back)

Figure 7.15: The RGB coloured pointcloud results obtained from observations of the rhinoceros pelvis scene using the Intel RealSense D435 with SEE and SEE++.

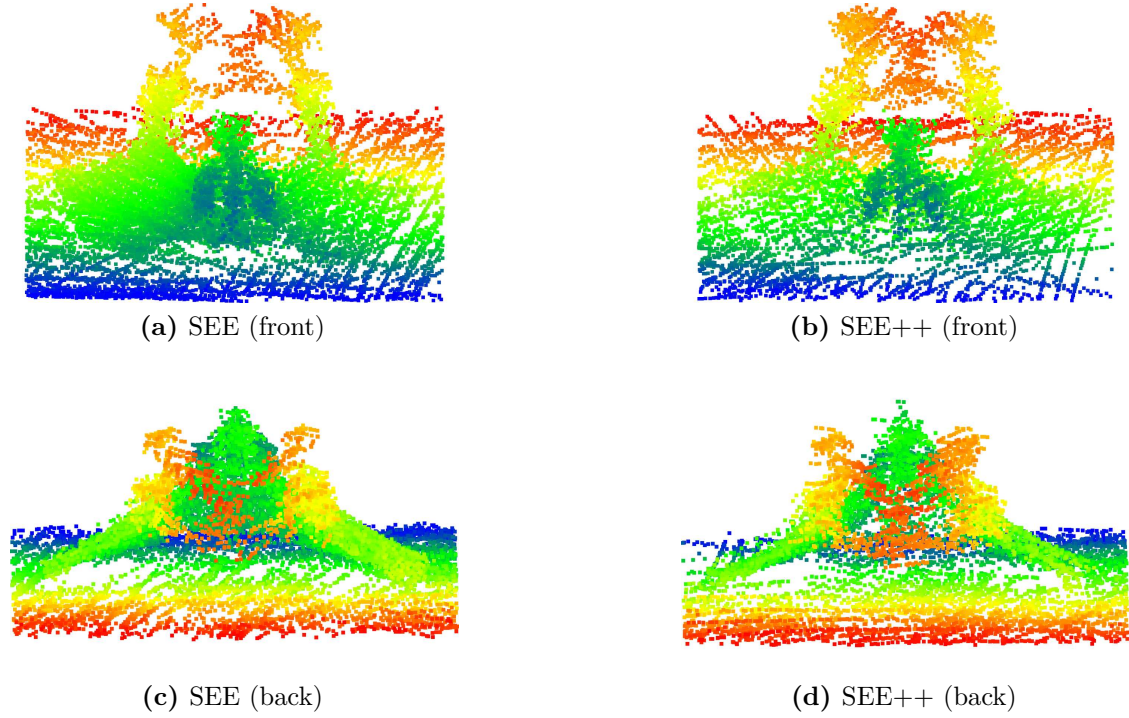


Figure 7.16: The y -coordinate coloured pointcloud results obtained from observations of the rhinoceros pelvis using the Velodyne VLP-16 with SEE and SEE++.

Sensors	RealSense		Velodyne	
Resolution (m), r	0.05		0.05	
Target Density, ρ	500000		20000	
Algorithms	SEE	SEE++	SEE	SEE++
Number of Views	68	58	39	27
Travel Distance (m)	24	20	19	15
Computation Time (s)	76	208	2	3

Table 7.5: Top: the resolution radius, r , and target measurement density, ρ , used for observations of the rhinoceros pelvis scene with the Intel RealSense D435 and the Velodyne VLP-16. Bottom: the number of views, travel distance and computation time required to obtain scene observations with SEE and SEE+ using each sensor.

obtained denser measurements of the illium (Fig. 7.16a) while SEE++ captured more measurements from the pubis and ischium (Fig. 7.16d). SEE++ likely required fewer views to obtain a sufficient measurement density on the large and relatively continuous surface of the illium as it considered the visibility of frontier points when selecting next best views.

The quantitative performance metrics demonstrate that SEE++ observed the scene using 31% fewer views and a 21% shorter travel distance than SEE. The computation time used by SEE++ was 1.5 times greater than required by SEE. Both SEE and SEE++ were able to observe this scene using fewer views, shorter travel distances and a lower computation time than the other scenes due to its smaller scale and the use of a smaller resolution radius.

7.2.6 Crocodile Skull

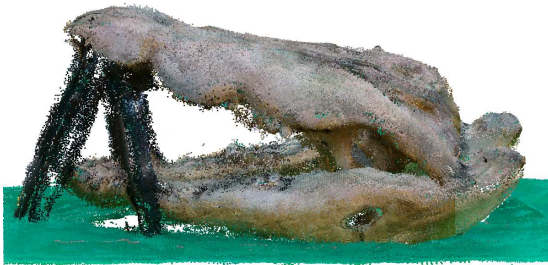
The crocodile skull scene (Fig. 7.17; Fig. A.26) contains the skull of a saltwater crocodile (*crocodylus porosus*) specimen that was loaned from the Oxford University Museum of Natural History (OUMNH 19149). The skull exhibits highly detailed visual features and texture. Its surface geometry is incredibly complex and contains numerous interior surfaces that are only partially visible when the jaw is closed. The desire to capture measurements from the interior surfaces of the crocodile skull (e.g., the jaw joints) motivated the decision to prop open the jaw with a tripod when observing the scene. This enabled the visibility of previously hidden surfaces without significantly reducing the complexity of self-occlusions in the scene.

7.2.6.1 RealSense

The RealSense observations were obtained using a target measurement density of 1000000 points per m^3 and a resolution of 0.01 m. The measurement density was increased and resolution radius was decreased with the aim of attaining highest observation quality obtainable with the RealSense. The qualitative pointcloud results (Fig. 7.18; Fig. A.27 and Fig. A.28) show that SEE++ was able to observe the scene structure with higher fidelity than SEE in many places (e.g., the nostril;



Figure 7.17: Photographs of the crocodile skull scene. It contains the skull of a saltwater crocodile (*crocodylus porosus*) specimen that was loaned from the Oxford University Museum of Natural History (OUMNH 19149).



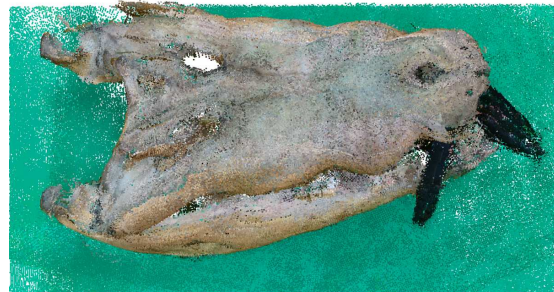
(a) SEE (side)



(b) SEE++ (side)



(c) SEE (top)



(d) SEE++ (top)

Figure 7.18: The RGB coloured pointcloud results obtained from observations of the crocodile skull scene using the Intel RealSense D435 with SEE and SEE++.

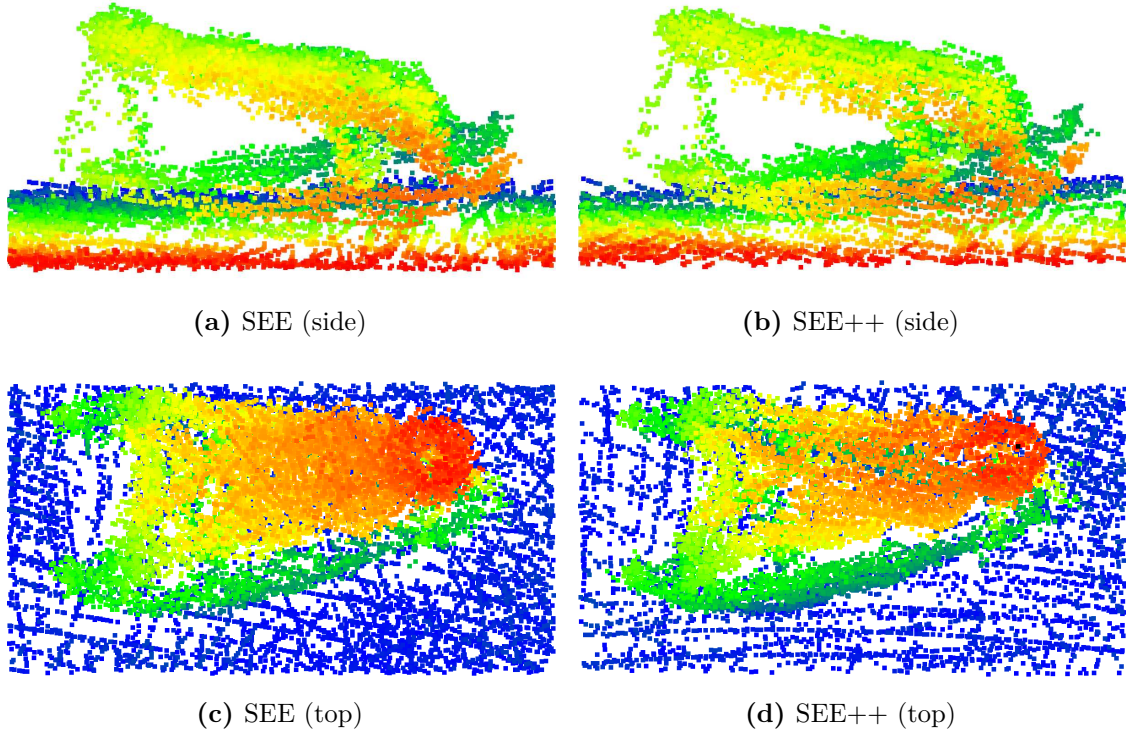


Figure 7.19: The y - and z -coordinate coloured pointcloud results obtained from observations of the crocodile skull using the Velodyne VLP-16 with SEE/SEE++.

Sensors	RealSense		Velodyne	
Resolution (m), r	0.01		0.05	
Target Density, ρ	1000000		20000	
Algorithms	SEE	SEE++	SEE	SEE++
Number of Views	226	134	48	34
Travel Distance (m)	85	46	29	23
Computation Time (s)	55	158	1	4

Table 7.6: Top: the resolution radius, r , and target measurement density, ρ , used for observations of the crocodile skull scene with the Intel RealSense D435 and the Velodyne VLP-16. Bottom: the number of views, travel distance and computation time required to obtain scene observations with SEE and SEE+ using each sensor.

Fig. 7.18d) but the measurements of details in the scene structure were often noisy for both approaches and certain surfaces could not be successfully observed.

SEE++ could not successfully observe the lefthand jaw joint (Fig. 7.18b) as the RealSense was unable to capture reliable measurements from this surface despite it being visible from multiple views. SEE was able to successfully observe this region from a view with an orientation orthogonal to the local surface geometry but none of the views captured by SEE++ of the region were orthogonal to the surface and this evidently precluded the capture of reliable measurements. The absence of an orthogonal view for SEE++ is likely a byproduct of decisions made by the optimisation strategy for identifying unoccluded views (e.g., as a result of self-occlusions between the jaw joint and the ground plane) and the consideration of scene visibility prioritising views that could also observe other regions of the scene while only obtaining measurements of the jaw joint from more acute angles.

The quantitative performance metrics (Table 7.6) demonstrate that SEE++ observed the scene using 41% fewer views and a 46% shorter travel distance than SEE. SEE++ required a 2.9 times greater computation time than SEE to obtain an observation. This illustrates the importance of proactively handling occlusions and considering scene visibility for enabling the efficient observation of a scene with complex surface geometry using a high measurement density. The relative increase in computation time is significant as the decreased resolution radius reduces the cost of updating the density representation for both approaches but for SEE++ this is offset by the cost of updating the covisibility graph and using the optimisation strategy to propose unoccluded views.

7.2.6.2 Velodyne

The Velodyne observations were obtained using a target measurement density of 20000 points per m^3 and a resolution of 0.05 m. As for the Rhinoceros Pelvis scene it was possible to use a smaller resolution radius due to the reduction in measurement noise. The qualitative pointcloud results (Fig. 7.19; Fig. A.29 and Fig. A.30) show that both SEE and SEE++ obtained similarly complete observations.

The quantitative performance metrics (Table 7.6) demonstrate that SEE++ observed the scene using 29% fewer views and a 21% shorter travel distance than SEE. The computation time used by SEE++ was four times greater than required by SEE. This improvement in observation performance illustrates that it is possible to achieve reductions in both the number of views and travel distance required to observe a small scene with complex surface geometry provided regions of visibility exist between self-occluding surfaces which allow them to be observed from unoccluded views that are close to the current sensor position.

7.2.7 Summary

The qualitative pointcloud results presented in the preceding subsections demonstrate that both SEE and SEE++ were able to capture largely complete observations of scenes with varying sizes and structural complexities independently using either an Intel RealSense D435 or a Velodyne VLP-16. These observations were successfully obtained despite the presence of measurement noise for both sensors. The magnitude of noise was particularly significant for the RealSense and varied with the texture of surfaces being observed, the distance of the sensor from scene surfaces and the orientations of views relative to the observed surface geometry. The noise associated with the Velodyne measurements did not vary appreciably with the distance and angle of views except for the measurements obtained from reflective surfaces (e.g., the numerical markings on the boxes), which exhibited significant sensor noise.

The filtering that was applied to measurements reduced the level of noise but erroneous points that were temporally consistent remained (e.g., noisy points in the floor region of the double towers scene; Fig. 7.9). In some cases the filtering resulted in certain surfaces not being observed when reliable measurements could not be obtained (e.g., the jaw joint in the crocodile skull scene; Fig. 7.18b).

The parameters used by SEE and SEE++ to observe the scenes with each sensor were chosen to robustly handle the magnitude of measurement noise. The resolution radius was set to be sufficiently large to account for the variance of noisy measurements. It was increased with the scene size for the RealSense observations

(i.e., from 0.05 m for the single box and single tower scenes to 0.1 m for the double towers scene) as the magnitude of measurement noise was greater for surfaces farther from the sensor. A larger resolution was used for the Velodyne observations of the box scenes than the specimen scenes (i.e., 0.1 m vs. 0.05 m) due to the presence of the highly noisy measurements obtained from the reflective numerical markings.

The target measurement density was chosen to be large enough that frontier points could be reliably identified along observation boundaries while accounting for the sensor resolution and magnitude of measurement noise. A larger target density was used for the RealSense observations than the Velodyne observations (e.g., 500000 vs. 20000 for the single box scene) due to the greater sensor resolution. It was necessary to decrease the target density when observing scenes using the RealSense that exhibited greater noise to try and prevent noisy points, which were offset from true surfaces, from being erroneously identified as frontiers (e.g., reducing the target density from 500000 for the single tower scene to 100000 for the double towers scene).

The pointcloud results obtained by SEE and SEE++ using the RealSense demonstrate that SEE was often more successful at obtaining reliable measurements from the edges of scene surfaces (e.g., the box edges in the single box scene; Fig. 7.3) as these measurements could typically only be captured from adjusted views with visibility of the surfaces on both sides of an edge. The views proposed and selected using SEE++ by considering occlusions and scene visibility meant that fewer view adjustments were required. These views were more likely to have an orientation from which the observation of a continuous surface could be improved rather than one from which both sides of a surface discontinuity could be reliably observed.

The pointcloud results obtained for observations of the box scenes using the Velodyne demonstrate that SEE++ was often more successful at obtaining reliable measurements of the reflective numerical markings than SEE (e.g., for the single tower scene; Fig. 7.7). This can be attributed to the proposal and selection of more views orthogonal to the box sides using SEE++ than those obtained by SEE.

These differences illustrate that when capturing observations of real world scenes the magnitude of sensor noise associated with the measurements obtained

varies significantly with the observed surface texture and the distance and relative orientation of views to the scene surfaces being observed. While in many cases the most reliable measurements are obtained from views with orthogonal orientations to visible surfaces it is clear that this does not hold true for all sensors and surface structures (e.g., the box edges). It may be possible to account for the variation in sensor noise for different sensors when filtering measurements and processing newly observed points with SEE and SEE++ by computing a model of measurement noise for each sensor, but that was beyond the scope of these real world experiments.

The presented results show the pointcloud observations of the scenes that were obtained independently using either an Intel RealSense D435 or a Velodyne VLP-16. These pointclouds provide the most veracious qualitative representation of the scene coverage attained by SEE and SEE++. However, to utilise the observations for other purposes it would typically be necessary to compute a surface mesh representation from the pointclouds (e.g., a Poisson reconstruction; Kazhdan et al. 2006).

7.3 Evaluation

The view distance used for the observations was computed with the method presented in Section 6.1.2. SEE++ used an occlusion search distance of $\psi = 1$ m and a view visibility update limit of $\tau = 100$ views. The performance of SEE and SEE++ is evaluated using the metrics defined in Section 3.3.4. Ground truth observations of the scenes could not be obtained so the surface coverage for these experiments is computed relative to the final pointcloud result obtained from an observation using a registration distance of $r_d = 0.005$ m. This is used to quantify the improvement in scene coverage achieved per view, unit of travel distance or unit of computation time.

A quantitative evaluation of the RealSense observations (Fig. 7.20 and Fig. 7.21) demonstrates that SEE++ was able to observe all of the scenes using fewer views and shorter travel distances than SEE while requiring a greater computational time. The number of views, travel distance and computation time required to obtain an observation typically increases with the scene size when the same target measurement density and resolution radius are used. The results demonstrate that

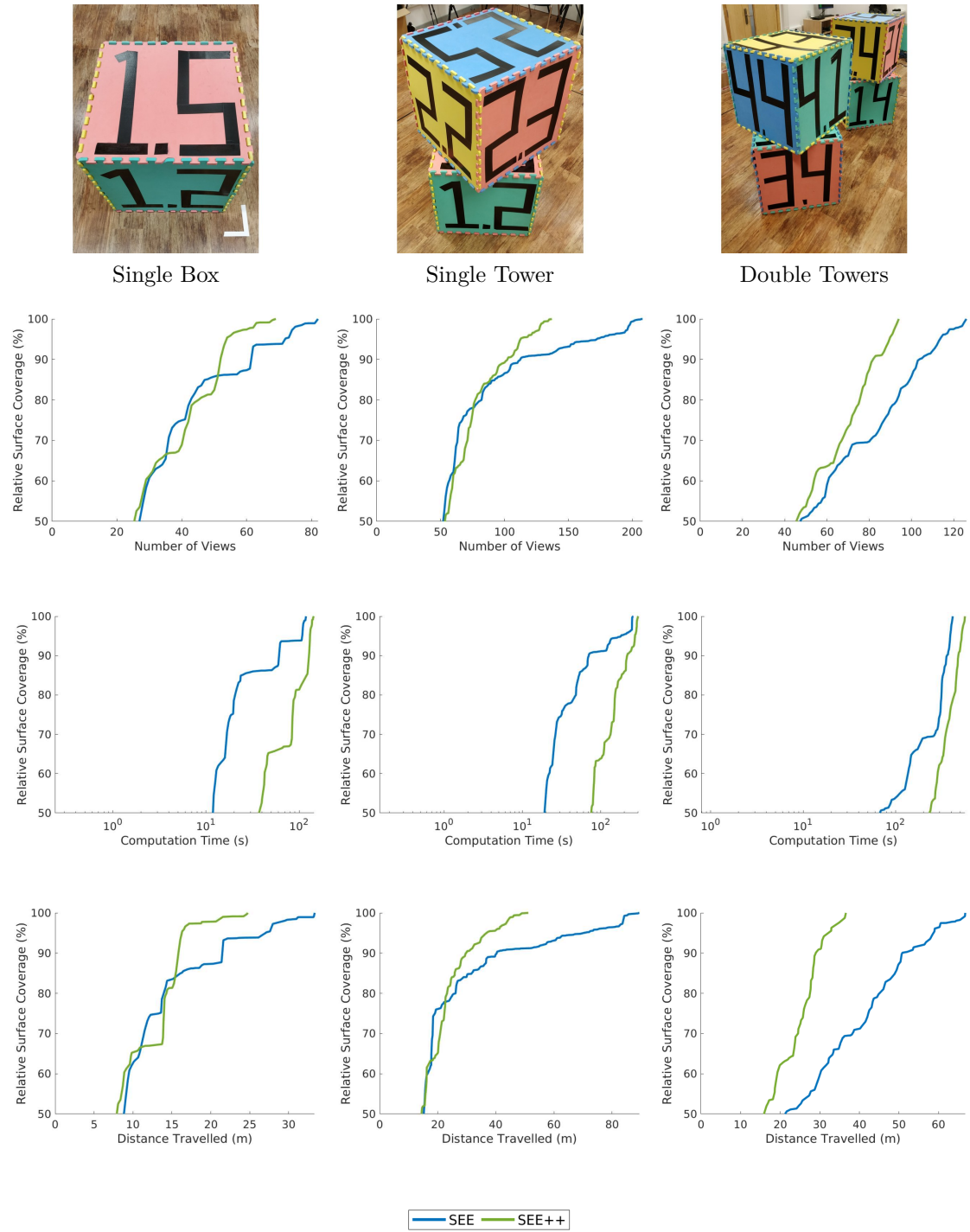


Figure 7.20: A quantitative evaluation of SEE and SEE++ for observations of the single box, single tower and double towers scenes using the Intel RealSense D435. The graphs present the relative surface coverage achieved by SEE and SEE++ with, from top to bottom, the number of views, computation time and travel distance.

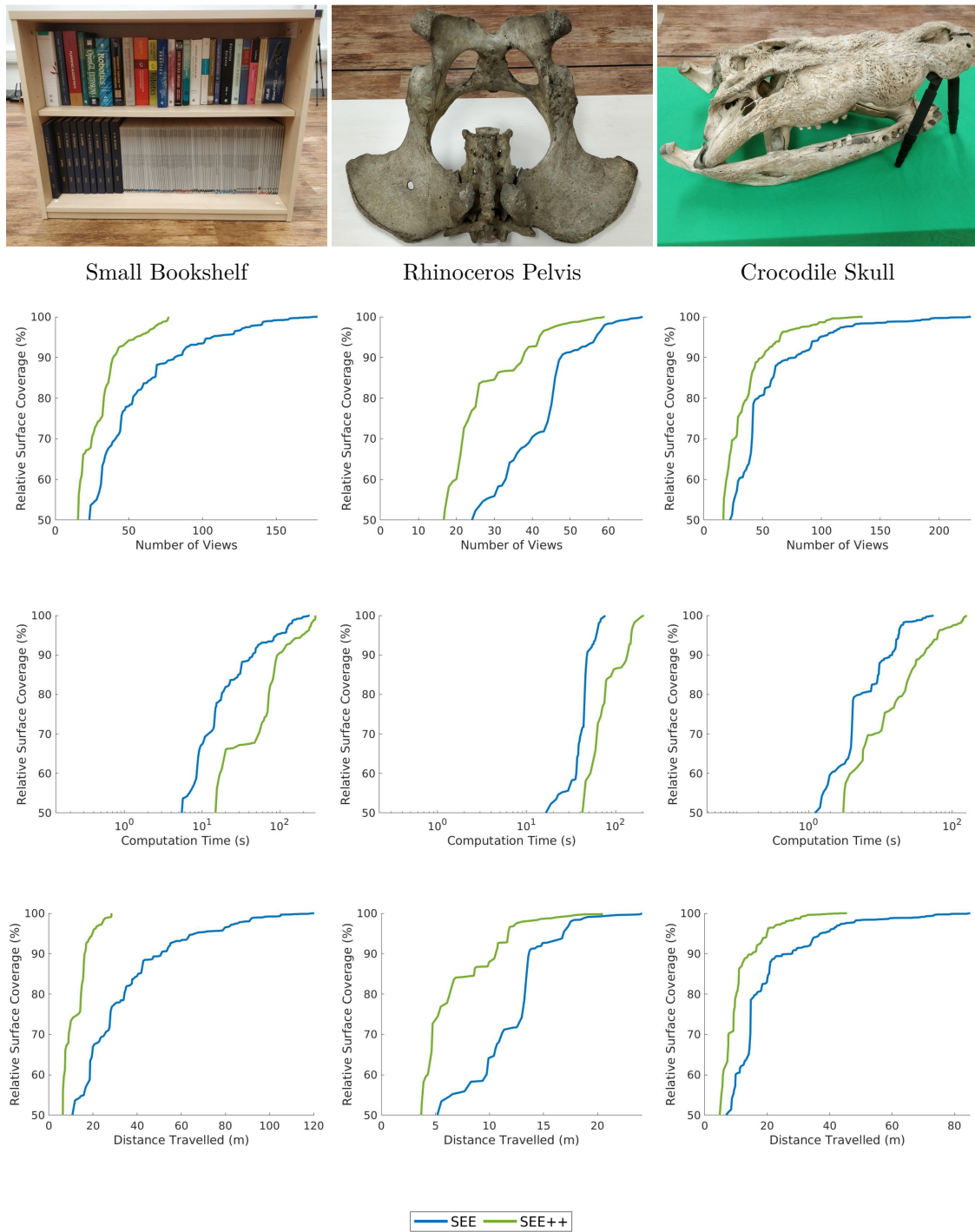


Figure 7.21: A quantitative evaluation of SEE and SEE++ for observations of the small bookshelf, rhinoceros pelvis and crocodile skull using the Intel RealSense D435. The graphs present the relative surface coverage achieved by SEE and SEE++ with, from top to bottom, the number of views, computation time and travel distance.

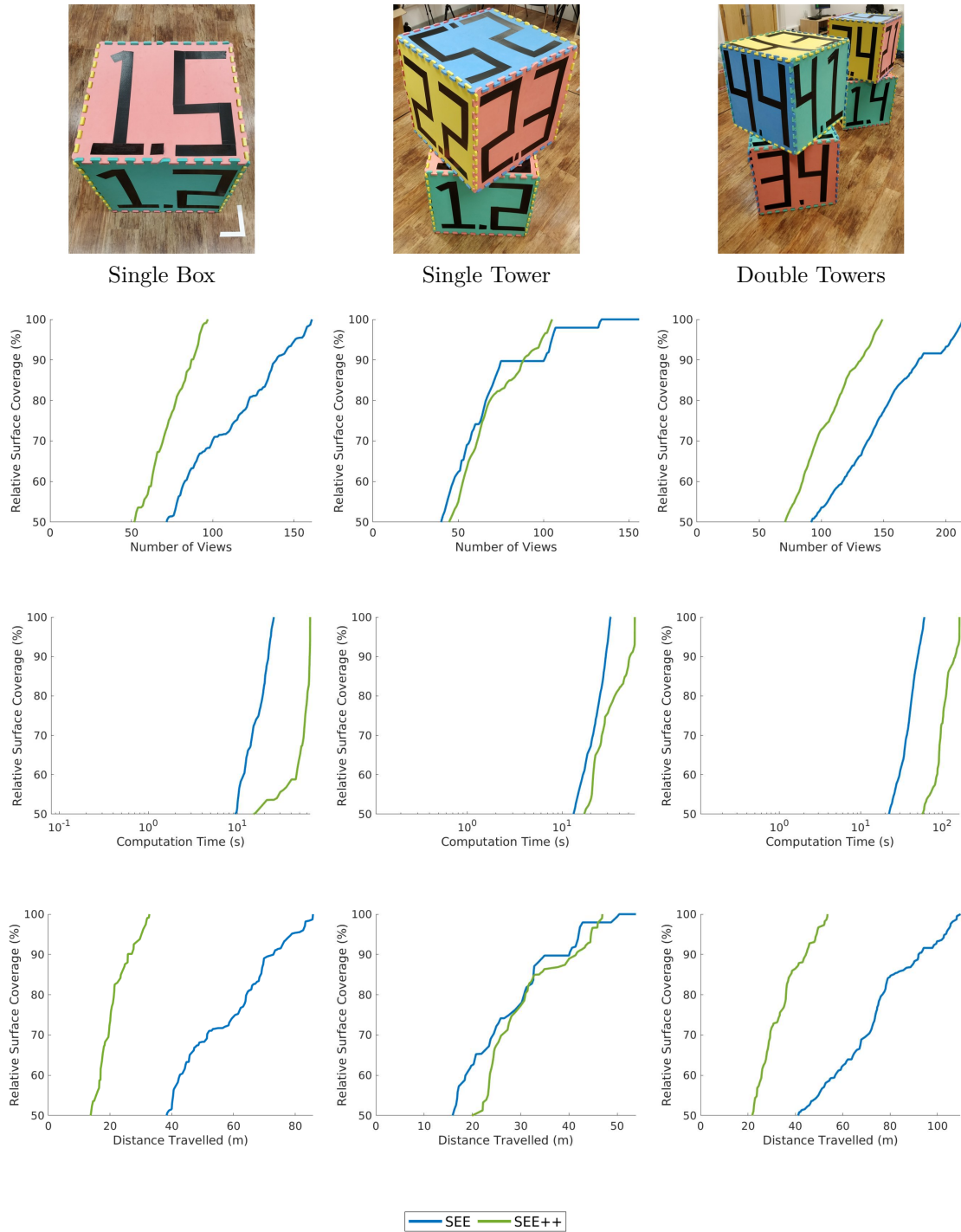


Figure 7.22: A quantitative evaluation of SEE and SEE++ for observations of the single box, single tower and double towers scenes using the Velodyne VLP-16. The graphs present the relative surface coverage achieved by SEE and SEE++ with, from top to bottom, the number of views, computation time and travel distance.

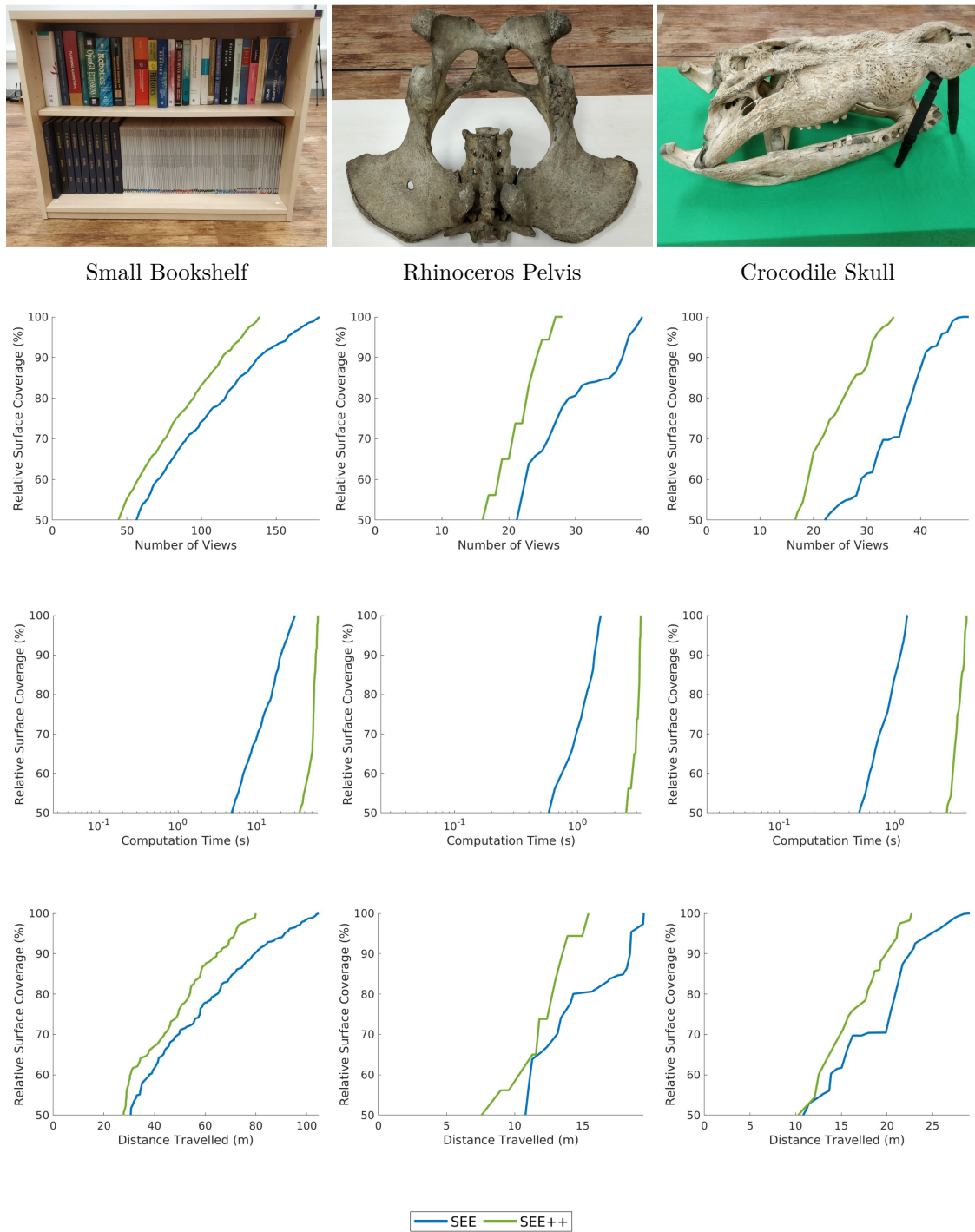


Figure 7.23: A quantitative evaluation of SEE and SEE++ for observations of the small bookshelf, rhinoceros pelvis and crocodile skull using the Velodyne VLP-16. The graphs present the relative surface coverage achieved by SEE and SEE++ with, from top to bottom, the number of views, computation time and travel distance.

the computational cost scales significantly better when the same parameters can be used to observe a larger scene (e.g., for the single box and the single tower scenes) as increasing the resolution radius means that a greater computational cost is incurred when updating the density representation (e.g., for the double towers scene).

The improved efficiency of SEE++ is evidenced by the relatively consistent and significant increases in surface coverage obtained from each view captured and per unit of distance travelled. This is illustrated in contrast to the improvements in surface coverage obtained by SEE which are nonexistent for certain sequences of views. These views can be identified in the graphs by horizontal segments in the plots. This typically occurs when attempting to observe a frontier point that is the product of sensor noise and is offset from a real surface which has already been fully observed. The effect is more prevalent for the box scenes due to their discontinuous surface geometry. When one side of a box has been fully observed the scene coverage will not be improved until a view is obtained of a partially observed side. The observation performance of SEE++ is less affected by this occurrence as when a frontier point is not successfully observed it can chosen an alternative view rather than applying incremental view adjustments that will likely also be unsuccessful at observing a frontier point that is the product of sensor noise.

A quantitative evaluation of the Velodyne observations (Fig. 7.22 and Fig. 7.23) demonstrates that SEE++ was able to observe all of the scenes using fewer views and shorter travel distances than SEE while requiring a greater computational time. The result graphs show that in most cases both SEE and SEE++ obtained relatively consistent improvements in surface coverage per view and unit of travel distance. However, the presence of horizontal segments in the plots for both approaches illustrate that they obtained some sequences of unsuccessful views which provided no improvements in surface coverage. This is particularly evident in the SEE plot on the number of views graph for the single tower scene as it shows three distinct sequences of unsuccessful views. This plot illustrates that the more frequent use of incremental view adjustments by SEE, while sometimes beneficial (e.g., for observing the box edges), can incur a significant increase in the number of views obtained.

7.4 Discussion

The real world experiments presented in this chapter demonstrate that both SEE and SEE++ were able to obtain largely complete observations of real world scenes with varying size and complexity. Experiments were performed using both a stereo camera and LiDAR sensor to demonstrate the capability of SEE and SEE++ to generalise between different sensing modalities. The qualitative pointcloud results show that largely complete observations were successfully obtained despite the presence of measurement noise. A quantitative evaluation of the observation performance of SEE and SEE++ evidences the improved efficiency of SEE++ as it consistently obtained scene observations using fewer views and shorter travel distances than SEE at the expense of incurring a greater computational cost.

The key challenges addressed when performing these experiments were obtaining reliable estimates of the sensor wand pose and accounting for the measurement noise produced by each sensor. The problem of pose estimation was solved by using a Vicon system to define a known coordinate frame for the scenes and provide robust tracking of the sensor wand. Measurement noise was reduced for both sensors by applying filtering to the point measurements before processing a pointcloud observation. SEE and SEE++ were able to successfully obtain largely complete scene observations despite the noise associated with the measurements captured from some surfaces.

In summary, the work presented in this chapter makes three key contributions:

1. Real world experiments demonstrating the observation capabilities of SEE and SEE++ on several real world scenes with varying sizes and structural complexities independently using either a stereo camera or LiDAR sensor.
2. Qualitative pointcloud results showing that both SEE and SEE++ were able to obtain largely complete observations despite the presence of sensor noise.
3. Quantitative metrics of the observation performance for SEE and SEE++ demonstrating that the efficiency improvements of SEE++ enabled it to observe all of the scenes using fewer views and with less travelling than SEE.

8

Conclusion

The ability to capture high-quality scene observations using 3D sensors is crucial for the analysis and imitation of real world structures in virtual environments. Applications range from scanning small household objects using consumer products to surveying large-scale outdoor scenes with industrial inspection systems (i.e., from *bunnies* to *buildings*). These observations provide greater utility if they capture a realistic representation of the real world that is both accurate and complete.

Observation accuracy is largely determined by the capabilities of the sensor used to obtain 3D measurements. The completeness of an observation is typically defined by considering the coverage of measurements obtained over visible scene surfaces. The most significant factor in determining the completeness of an observation is the selection of views from which measurements of a scene are obtained. Selecting the ‘next’ view of a scene to obtain that will provide the ‘best’ improvement in a scene observation is known as the Next Best View (NBV) planning problem.

This thesis presents work on NBV planning using a novel unstructured density representation. In contrast to existing literature on the NBV problem (Ch. 2), which typically utilises structured volumetric or surface representations, this representation does not impose an external structure on sensor measurements. Observed points are used to directly represent information on the structure of a scene rather than being aggregated into voxels (i.e., for volumetric representations) or a triangulated

mesh (i.e., for surface representations). The fidelity of scene knowledge encoded by this representation is not limited by the resolution of an external structure and it is not necessary to impose any assumptions about the scene geometry on the observation. Potential next best views are proposed and selected by directly considering information on the scene structure that is encoded in observed points rather than being sampled *a priori* without accounting for the scene geometry. Views are obtained until a given measurement density is attained for all observable surfaces rather than terminating an observation after a fixed number of views.

The Surface Edge Explorer (SEE) is a novel NBV planning approach that utilises this unstructured density representation (Ch. 3). Observations are obtained by capturing measurements with a given minimum density from all observable scene surfaces. Observed points are classified based on the number of neighbouring measurements within a given resolution radius to identify a frontier between fully and partially observed scene regions. Views are proposed to extend the coverage of a scene observation by estimating the local surface geometry around frontier points. Next best views are selected from the set of proposed views to improve the scene observation while moving short distances. When an improvement in the surface coverage around a frontier point is not attained, its associated view is adjusted by considering newly observed points. An observation is considered complete when the target measurement density has been attained for all observable scene surfaces.

The work on SEE discussed in this thesis was first presented at the 2017 Joint Industry and Robotics CDTs Symposium and extended at the 2018 IEEE International Conference on Robotics and Automation:

Rowan Border, Jonathan D. Gammell, and Paul Newman (2017). “Inferring Surface Geometry from Point Clouds for Next Best View Planning”. In: *Joint Industry and Robotics CDTs Symposium*, pp. 1–2

Rowan Border, Jonathan D. Gammell, and Paul Newman (2018). “Surface Edge Explorer (SEE): Planning Next Best Views Directly from 3D Observations”. In: *IEEE International Conference on Robotics and Automation*, pp. 1–8

An experimental comparison of the observation performance of SEE with state-of-the-art volumetric approaches demonstrates that SEE is capable of obtaining

highly complete scene observations using fewer views and a significantly lower computational time. These observations are obtained using shorter travel distances than many of the volumetric approaches with the exception of those that consider occlusions. This motivated the investigation of novel point-based methods for considering occlusions and scene visibility with an unstructured scene representation.

Existing literature on NBV planning, including the volumetric approaches evaluated for comparison in this thesis, has illustrated the value of considering occlusions and scene visibility when proposing and selecting next best views. The techniques applied by approaches with structured representations typically utilise raycasting, which precludes their use with an unstructured representation. In order to consider occlusions and scene visibility with an unstructured representation it was necessary to investigate novel solutions that provided pointwise considerations.

The investigation of proactive occlusion handling (Ch. 4) presents a novel method for representing pointwise occlusions and several strategies that aim to utilise this representation to propose unoccluded views of target frontier points. An experimental evaluation of these strategies in comparison with SEE demonstrates that proposing and selecting views which are more likely to be unoccluded improves the efficiency of scene observations as fewer views and shorter travel distances are required. A statistical analysis of the observation performance for each strategy shows that the optimisation strategy is the most successful at proposing unoccluded views.

The investigation into considering scene visibility when selecting next best views (Ch. 5) presents a novel graphical representation for encoding the shared visibility of frontier points from the set of view proposals. Several metrics were investigated for selecting next best views from this covisibility graph with the aim of obtaining the greatest improvement in surface coverage while reducing the sensor travel distance. Experiments comparing the observation performance of these metrics with SEE demonstrate improvements in the efficiency of scene observations for each metric and illustrate the trade-off that exists between a reduction in the number of views required to observe a scene and reducing the overall sensor travel distance. A

statistical analysis of the increases in surface coverage and travel distance per view for each metric shows that the best overall performance is achieved by the EMR metric.

The best performing solutions to the challenges of considering occlusions and scene visibility with an unstructured representation were integrated with SEE to create SEE++ (Ch. 6). An experimental comparison of SEE++ with SEE and state-of-the-art volumetric approaches demonstrates that SEE++ achieves significant reductions in both the number of views and travel distance required to obtain scene observations while maintaining a reasonable computation time. This illustrates the importance of obtaining unoccluded views of scene surfaces and considering the improvement in surface coverage attainable when selecting next best views.

This work on SEE++ was presented at the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems:

Rowan Border and Jonathan D. Gammell (2020). “Proactive Estimation of Occlusions and Scene Coverage for Planning Next Best Views in an Unstructured Representation”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–8

The observation capabilities of SEE and SEE++ are shown to successfully transfer from the observation of scene models in a simulation environment to the observation of real world scenes (Ch. 7). Experimental results demonstrate that both approaches were able to successfully observe scenes with varying sizes and structural complexities independently using multiple sensing modalities. Qualitative pointcloud results show that the observations obtained were largely complete despite the presence of significant measurement noise. Quantitative evaluation metrics evidence the improvements in observation performance achieved by SEE++ over SEE. This work on demonstrating the real world observation capabilities of SEE and SEE++ is being prepared for submission to a field robotics journal.

The presence of sensor noise and the absence of measurements on some unobservable surfaces in the pointcloud results indicates that improvements in the quality of real world scene observations could be attained by accounting for sensor-specific variations in measurement noise. This could be achieved by computing a noise model for each sensor which captures the variation of measurement noise with

the surface textures being observed, the distance of the sensor from visible scene surfaces and the orientation of views relative to the surface geometry. This noise model would then be used to inform the filtering of sensor measurements and the proposal of potential next best views. Information on the existence of scene surfaces for which reliable measurements were not obtained from a given view could also be considered when proposing views. This would be achieved by including knowledge of noisy measurements that were removed by filtering in the scene representation.

During the course of this work it has become evident that providing publically available implementations of presented NBV planning approaches represents a valuable contribution to the research community. This enables researchers to pursue further developments of the work and evaluate comparisons with other approaches without incurring the overhead and uncertainty associated with producing a bespoke implementation. Open-source implementations of SEE and SEE++ will be made available in an upcoming journal paper in service of making such a contribution. This will hopefully encourage the pursuit of further work on NBV planning with an unstructured scene representation. Implementations of the presented novel point-based techniques for considering occlusions and scene visibility without using raycasting also have the potential to be applied in other research areas.

8.0.1 Contributions

In summary, this thesis makes the following key contributions:

1. A novel unstructured scene representation using measurement density, which is founded on the principle that obtaining a minimum measurement density on all scene surfaces is a sufficient condition to achieve a complete observation.
2. SEE, a NBV planning approach using this novel representation that imposes no assumptions on the scene structure. Views are proposed, selected and adapted to improve a scene observation by directly considering point measurements.
3. An investigation of point-based strategies for proactively handling occlusions which detect occluded views of target points and aim to propose alternative unoccluded views from which the target surfaces can be successfully observed.

4. An investigation of methods for considering scene visibility with an unstructured representation which aim to select next best views that can provide the greatest improvements in scene coverage while travelling short distances.
5. SEE++, a NBV planning approach that integrates the most successful methods for handling occlusions and considering scene visibility with SEE to greatly improve observation performance by utilising an increased computation time.
6. Real world demonstrations of the presented approaches obtaining observations of several different scenes using both a stereo camera and LiDAR sensor.
7. Implementations of SEE and SEE++ that will be made available open-source to aid further research into NBV planning with unstructured representations.

8.0.2 Future Work

The demonstrated success of using an unstructured density representation for improving the performance of scene observations illustrates that the use of unstructured representations for NBV planning is worthy of further investigation. Such research could consider extending the density representation presented in this thesis (e.g., varying the target density with the surface complexity) or investigating new unstructured representations (e.g., one that considers the distribution of points).

The formulation of novel strategies for proactively handling point-based occlusions provides an opportunity to extend the NBV planning literature on occlusion handling beyond approaches that rely on raycasting. Similar strategies could also be applied to other fields where occlusions are considered, such as for mapping and navigation tasks. The use of a covisibility graph to represent the shared visibility of target manifolds between views could be adapted for NBV approaches with volumetric and surface representations to improve their observation performance.

Appendices



Real World Observations

This appendix presents a complete set of photographs and pointcloud results, shown from multiple viewpoints, for each of the real world scenes discussed in Chapter 7.

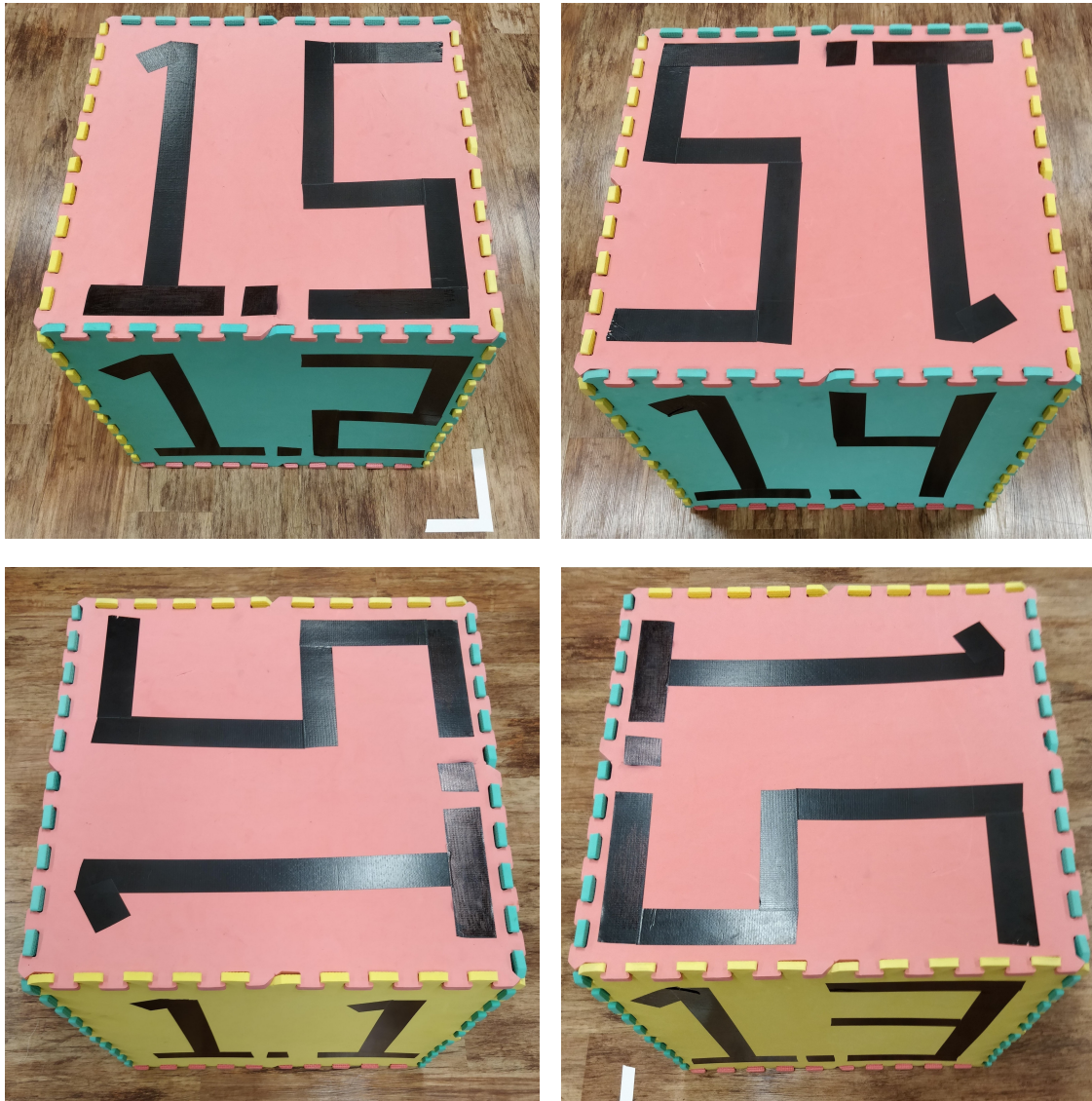


Figure A.1: Photographs of the single box scene. It contains a 0.6x0.6x0.6m foam box with a unique numerical marking on each side in reflective black duct tape.

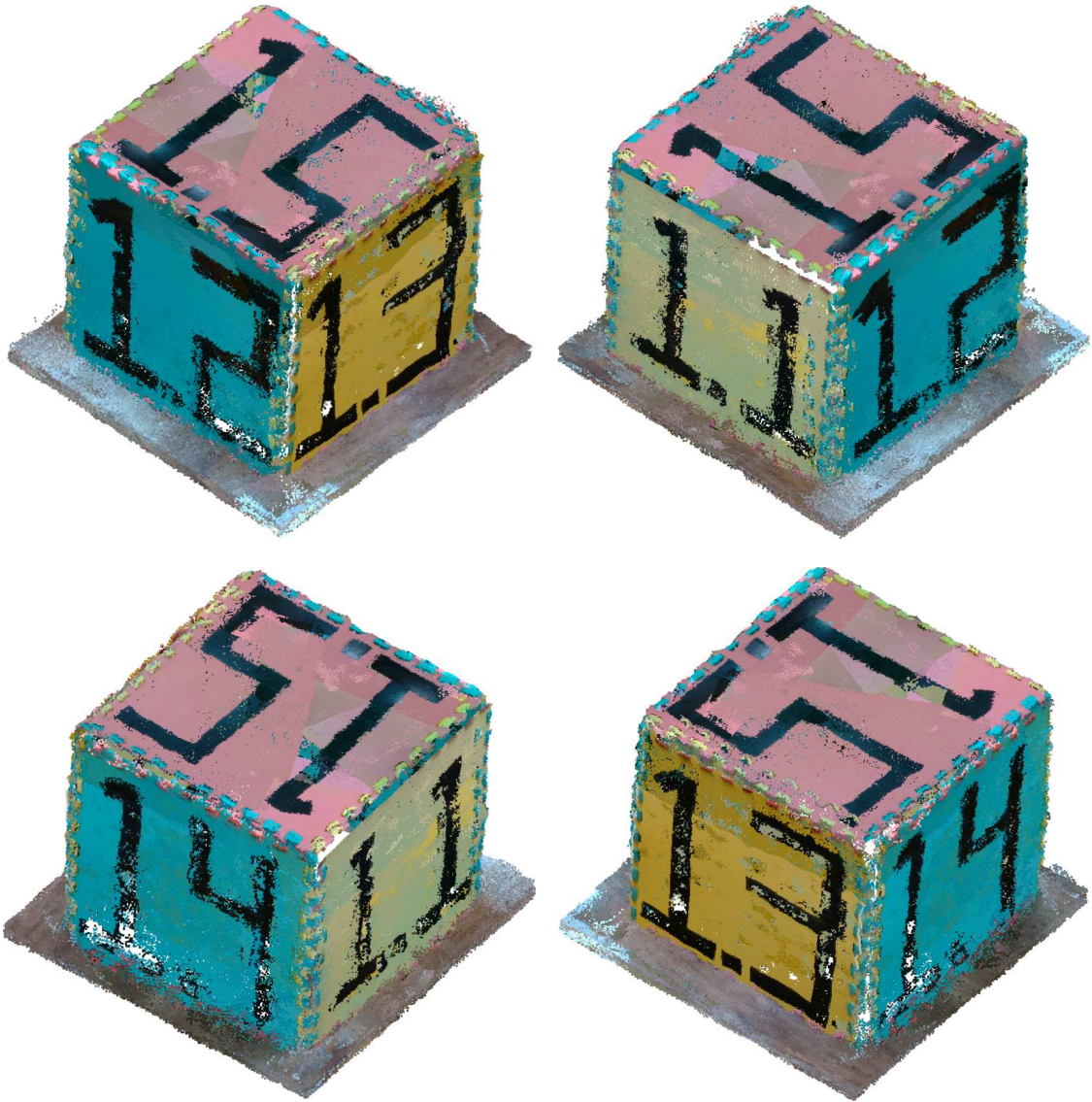


Figure A.2: The RGB coloured pointcloud result obtained from an observation of the single box scene using SEE with the Intel RealSense D435.

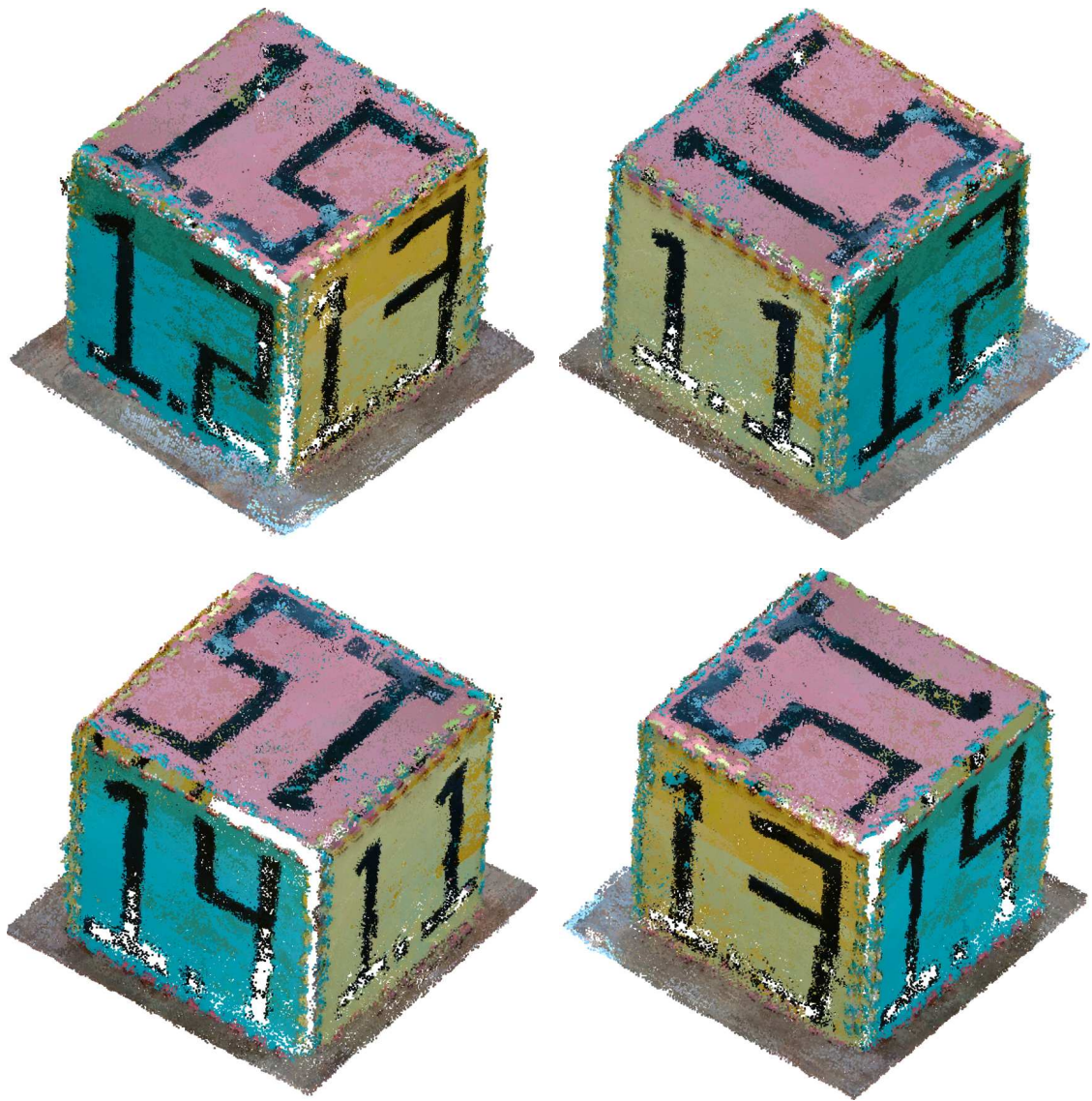


Figure A.3: The RGB coloured pointcloud result obtained from an observation of the single box scene using SEE++ with the Intel RealSense D435.

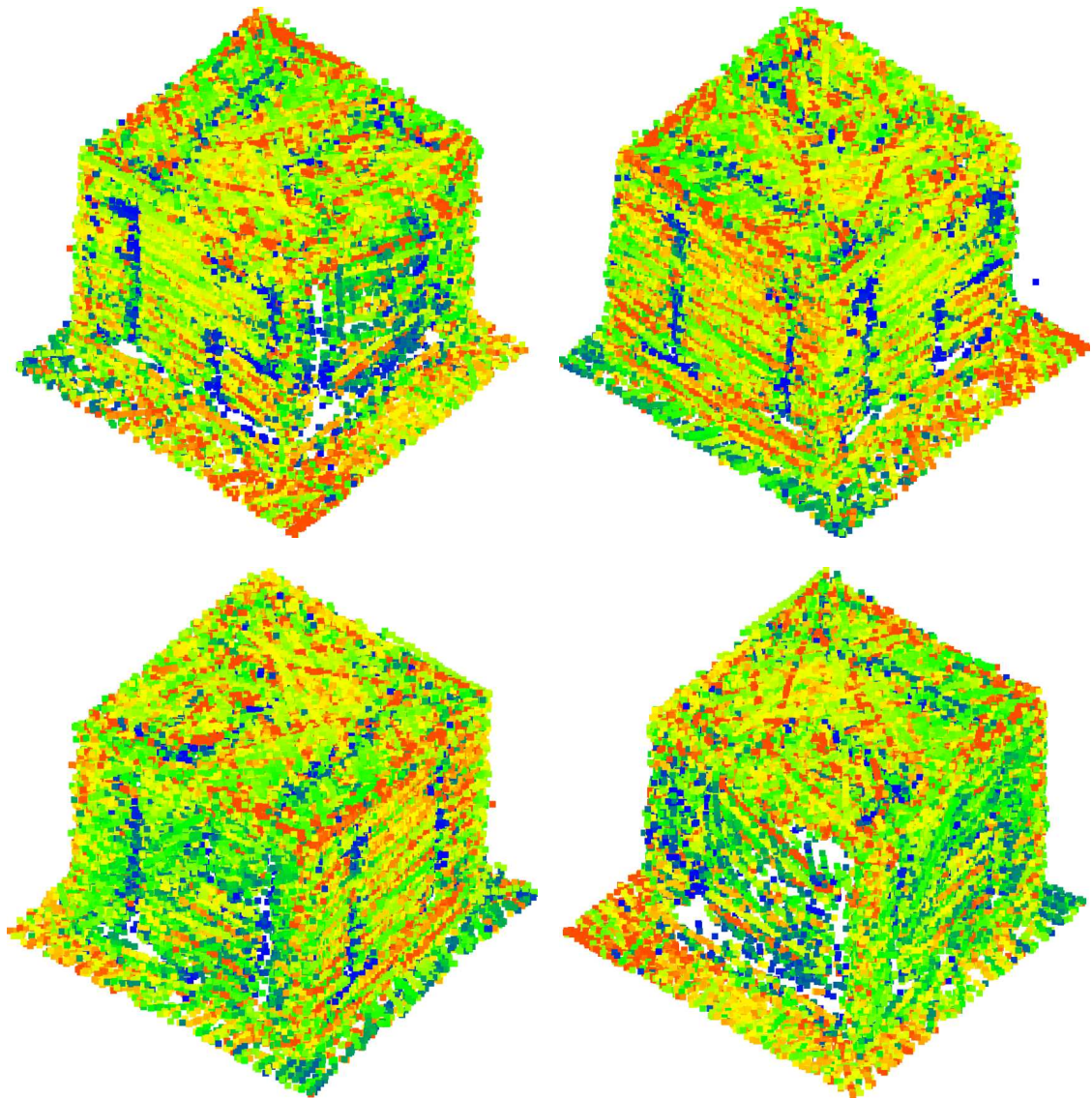


Figure A.4: The pointcloud result obtained from an observation of the single box scene using SEE with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each corner of the scene.

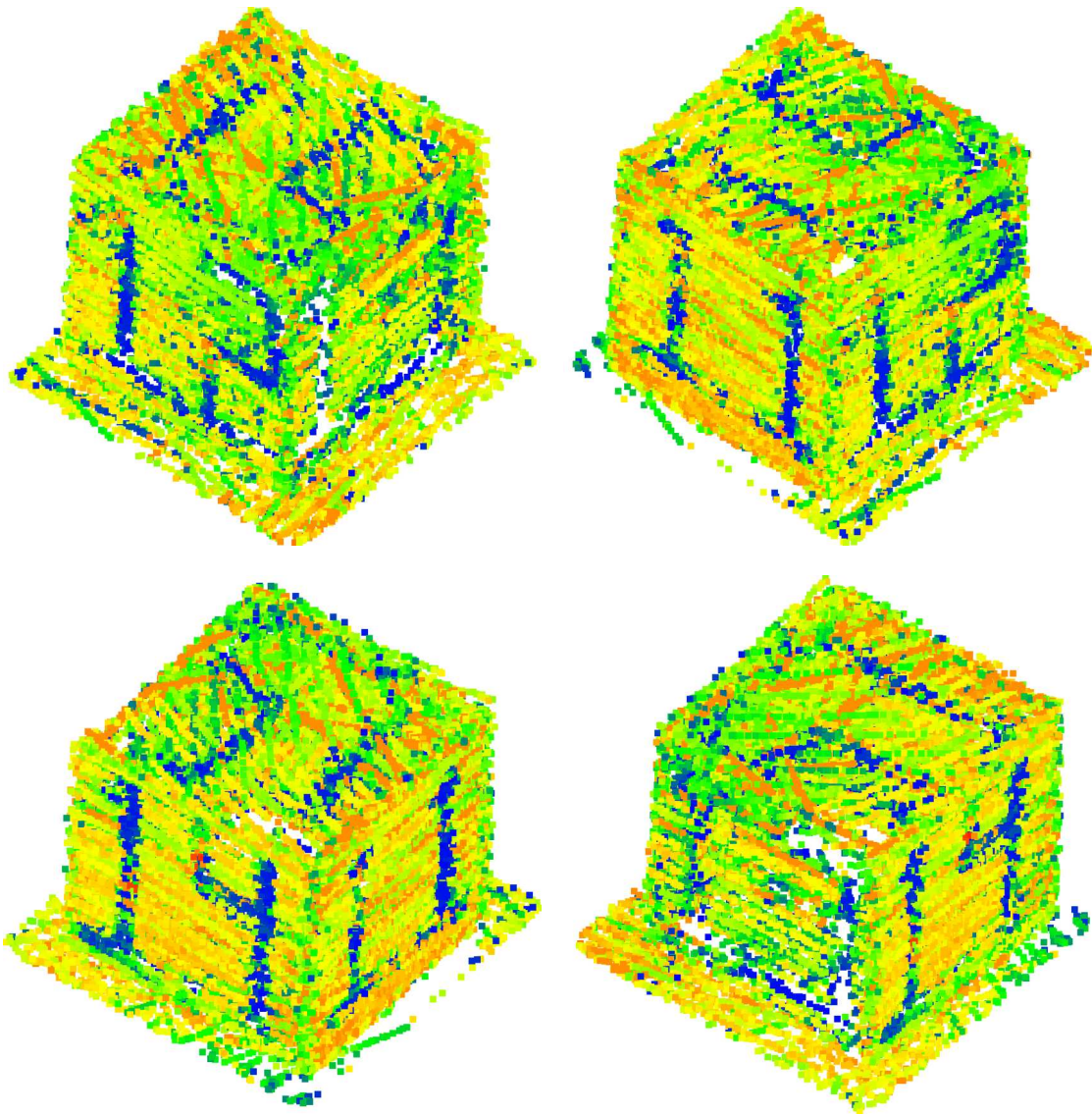


Figure A.5: The pointcloud result obtained from an observation of the single box scene using SEE++ with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each corner of the scene.

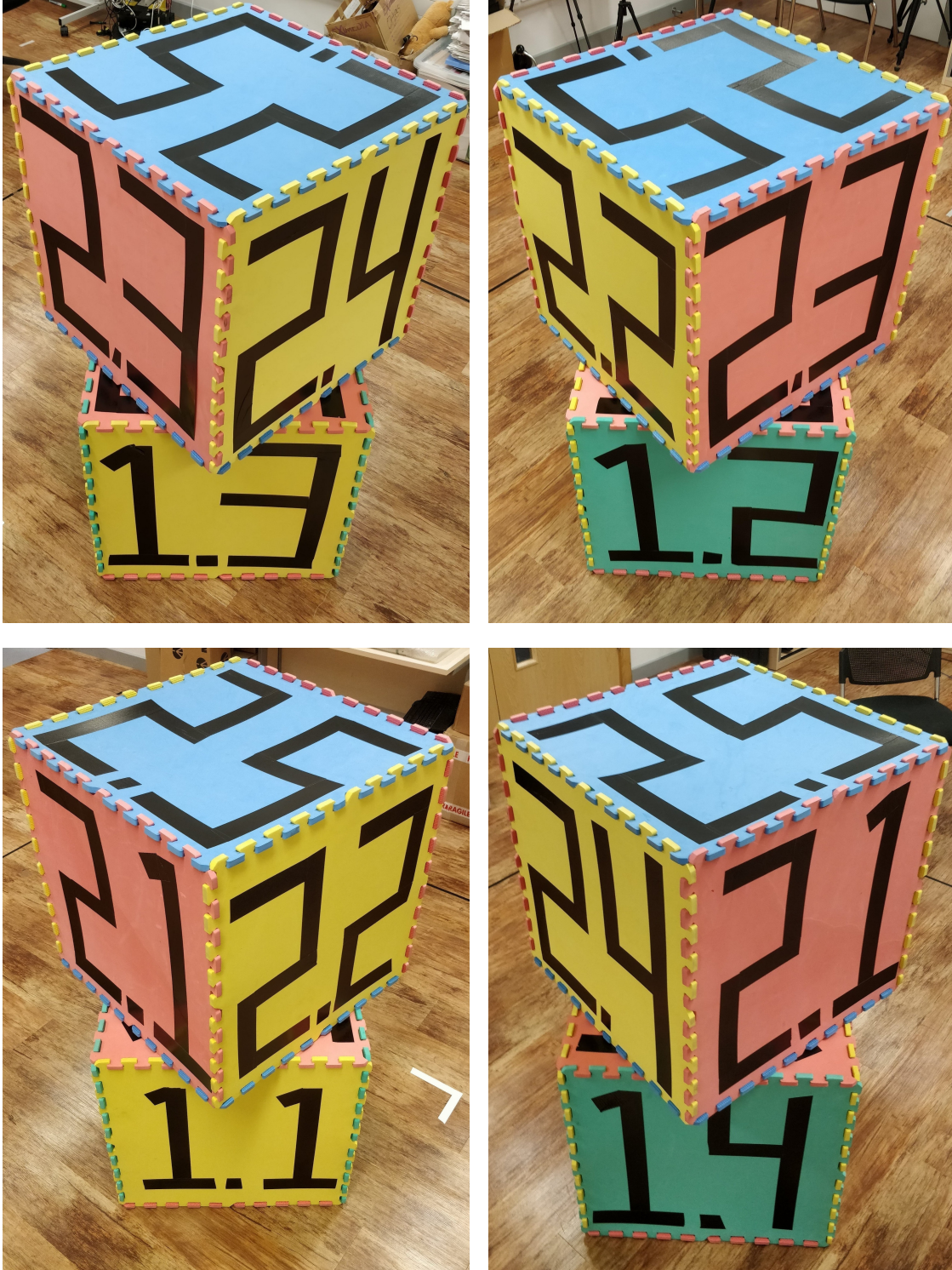


Figure A.6: Photographs of each side of the single tower scene. It contains two foam boxes in a stacked configuration with a 45° rotational offset between the boxes.

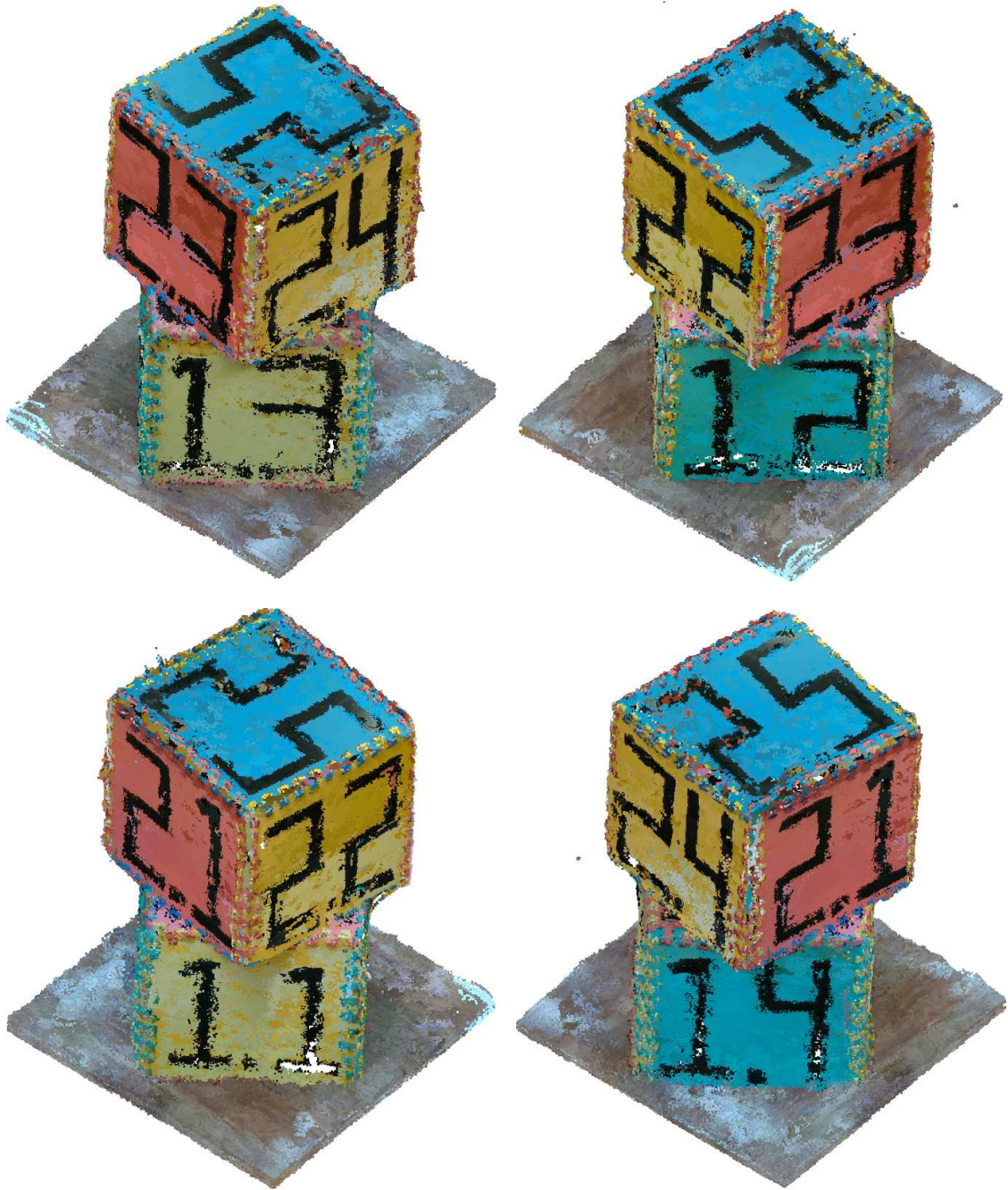


Figure A.7: The RGB coloured pointcloud result obtained from an observation of the single tower scene using SEE with the Intel RealSense D435.

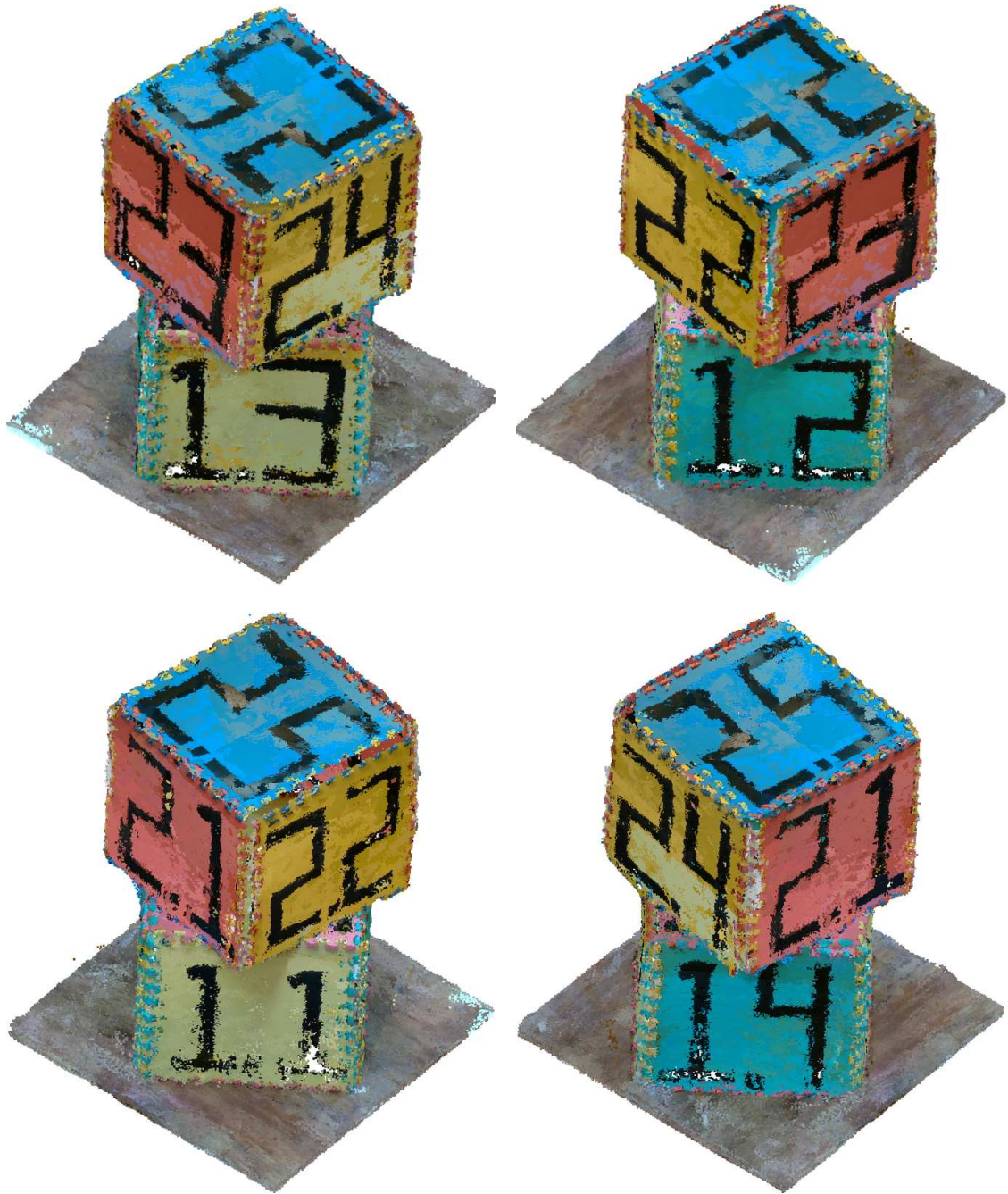


Figure A.8: The RGB coloured pointcloud result obtained from an observation of the single tower scene using SEE++ with the Intel RealSense D435.

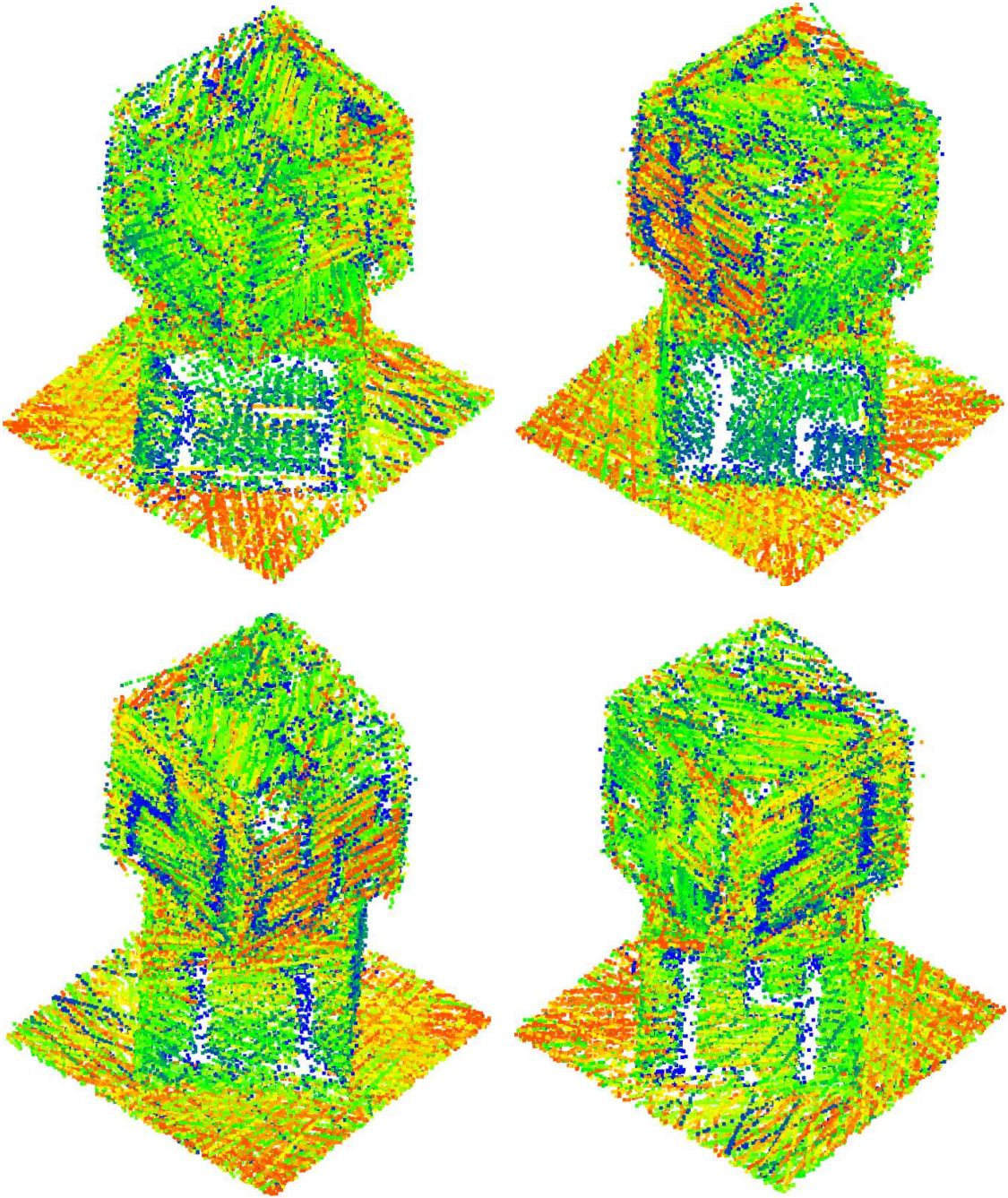


Figure A.9: The pointcloud result obtained from an observation of the single tower scene using SEE with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each corner of the scene.

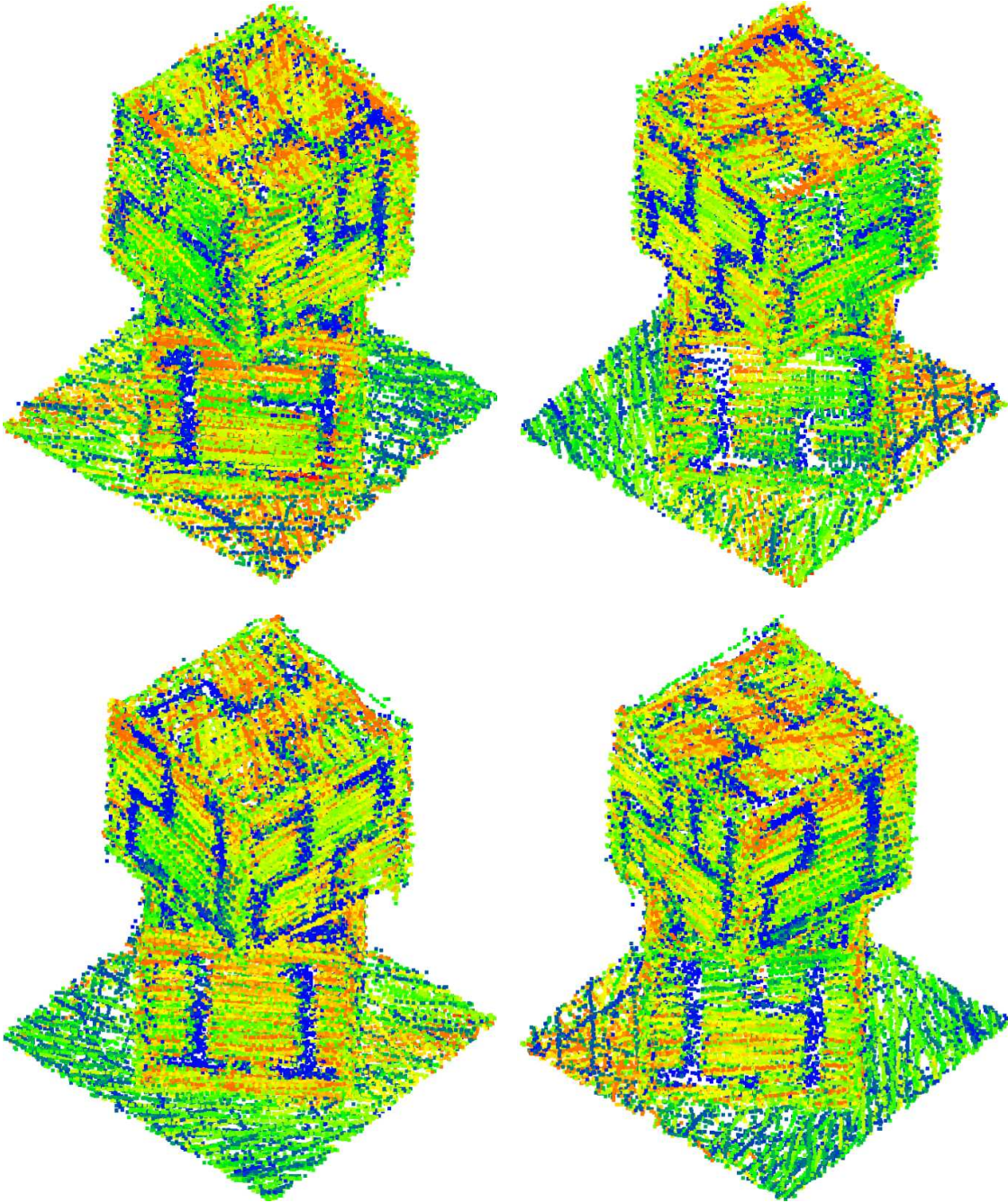


Figure A.10: The pointcloud result obtained from an observation of the single tower scene using SEE++ with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each corner of the scene.



Figure A.11: Photographs of each side of the double towers scene. It contains two stacks of rotationally offset foam boxes separated by a distance of less than 1 m.



Figure A.12: The RGB coloured pointcloud result obtained from an observation of the double towers scene using SEE with the Intel RealSense D435.

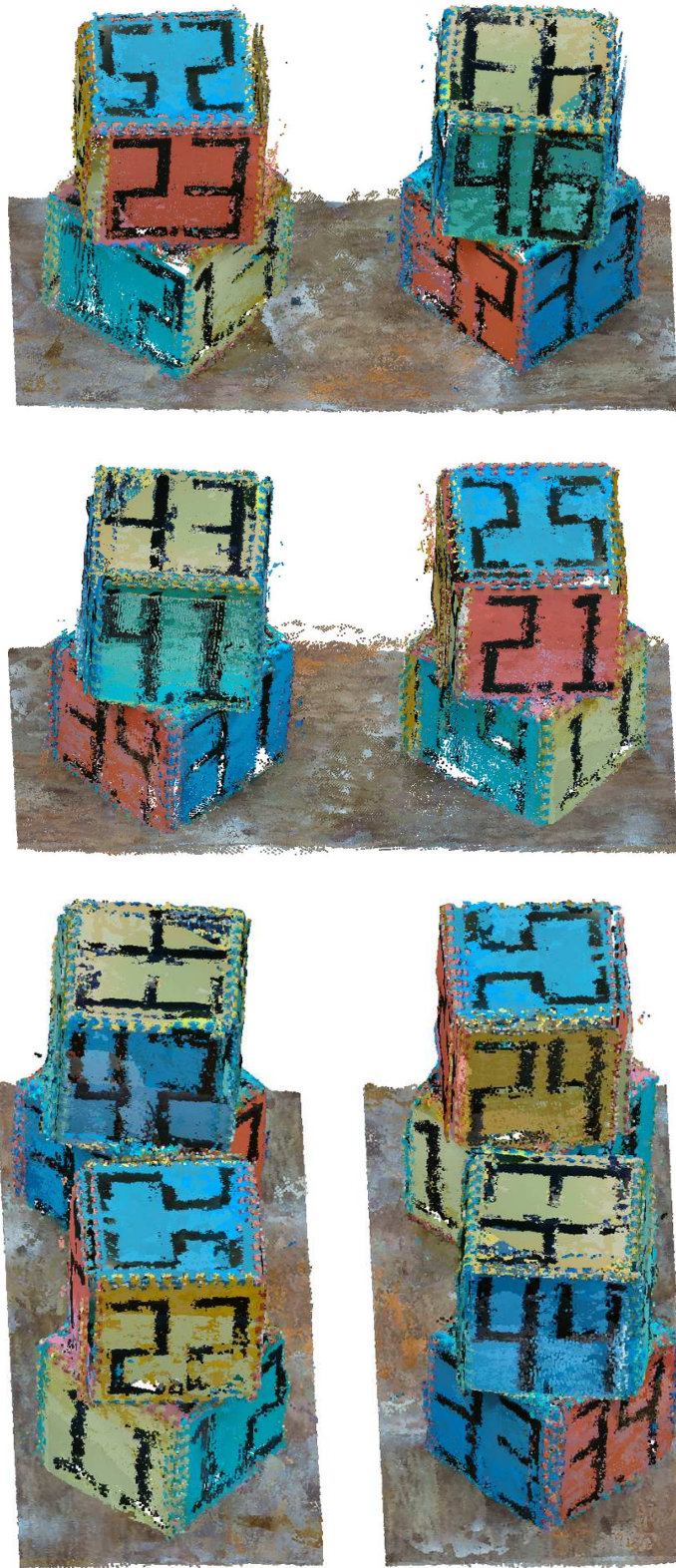


Figure A.13: The RGB coloured pointcloud result obtained from an observation of the double towers scene using SEE++ with the Intel RealSense D435.

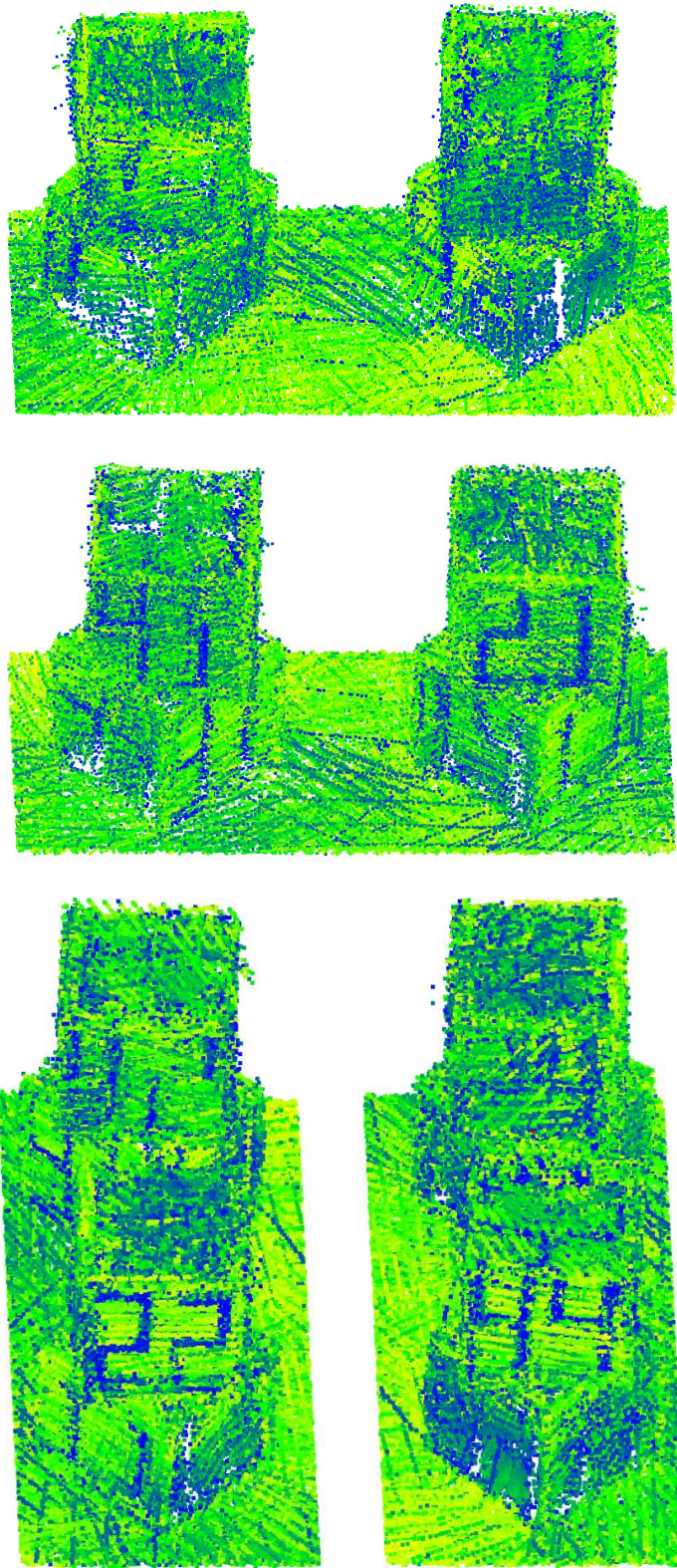


Figure A.14: The pointcloud result obtained from an observation of the double towers scene using SEE with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each side of the scene.

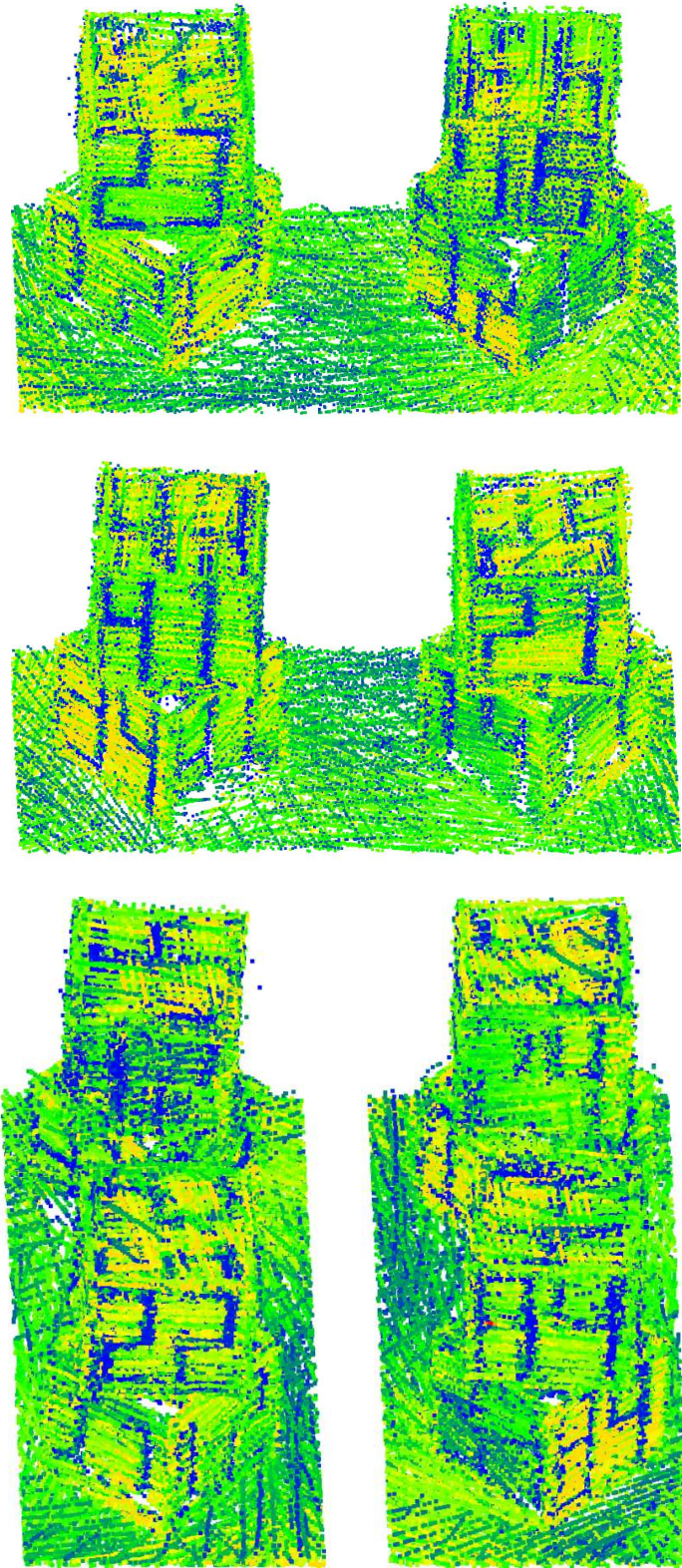


Figure A.15: The pointcloud result obtained from an observation of the double towers scene using SEE++ with the Velodyne VLP-16. The images use a colourmap based on the LiDAR intensity values and show views from each side of the scene.

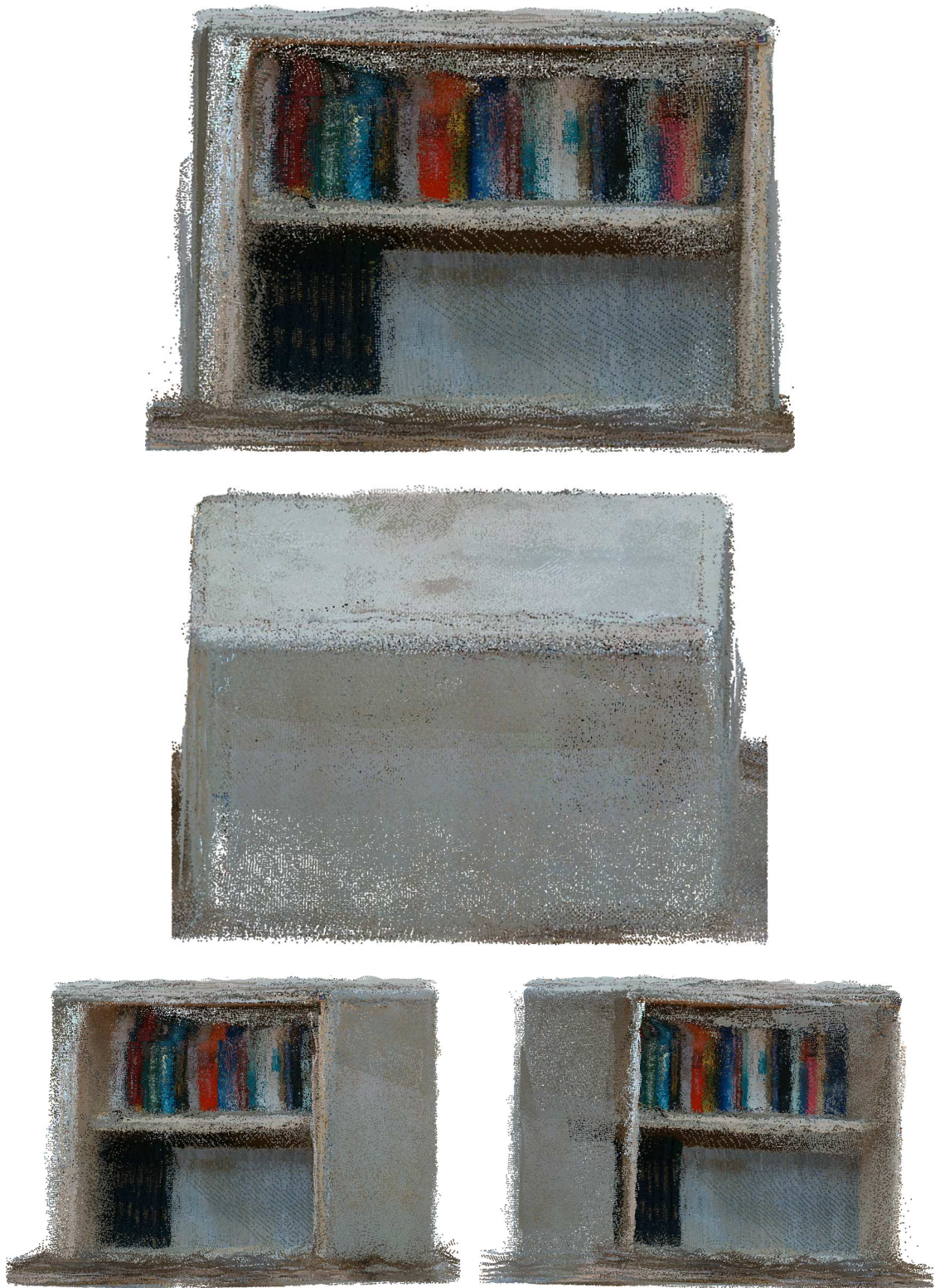


Figure A.17: The RGB coloured pointcloud result obtained from an observation of the small bookshelf scene using SEE with the Intel RealSense D435.



Figure A.18: The RGB coloured pointcloud result obtained from an observation of the small bookshelf scene using SEE++ with the Intel RealSense D435.

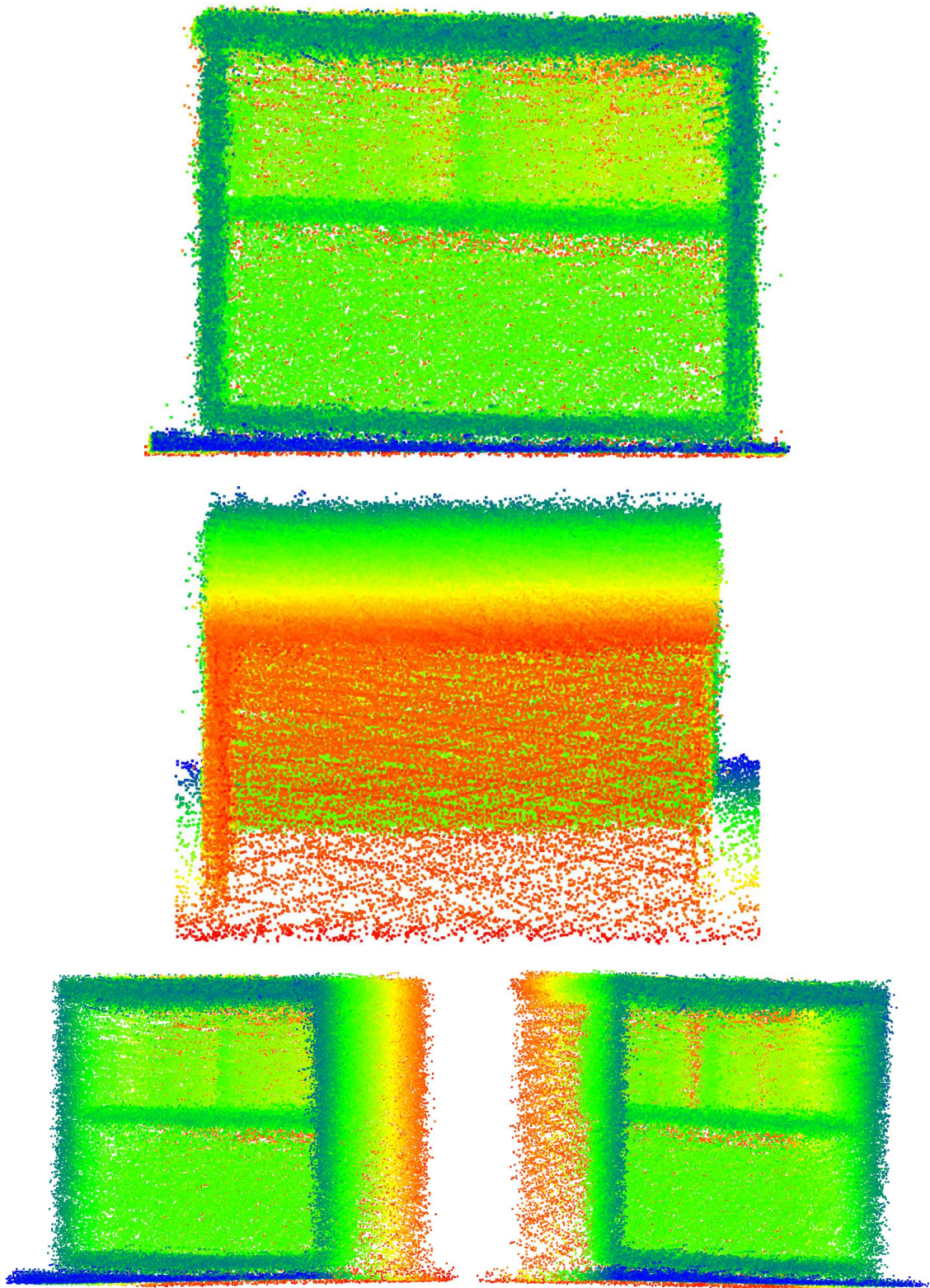


Figure A.19: The pointcloud result obtained from an observation of the small bookshelf scene using SEE with the Velodyne VLP-16. The images use a y -coordinate colourmap and show views of the front, back and two side angles of the bookshelf.

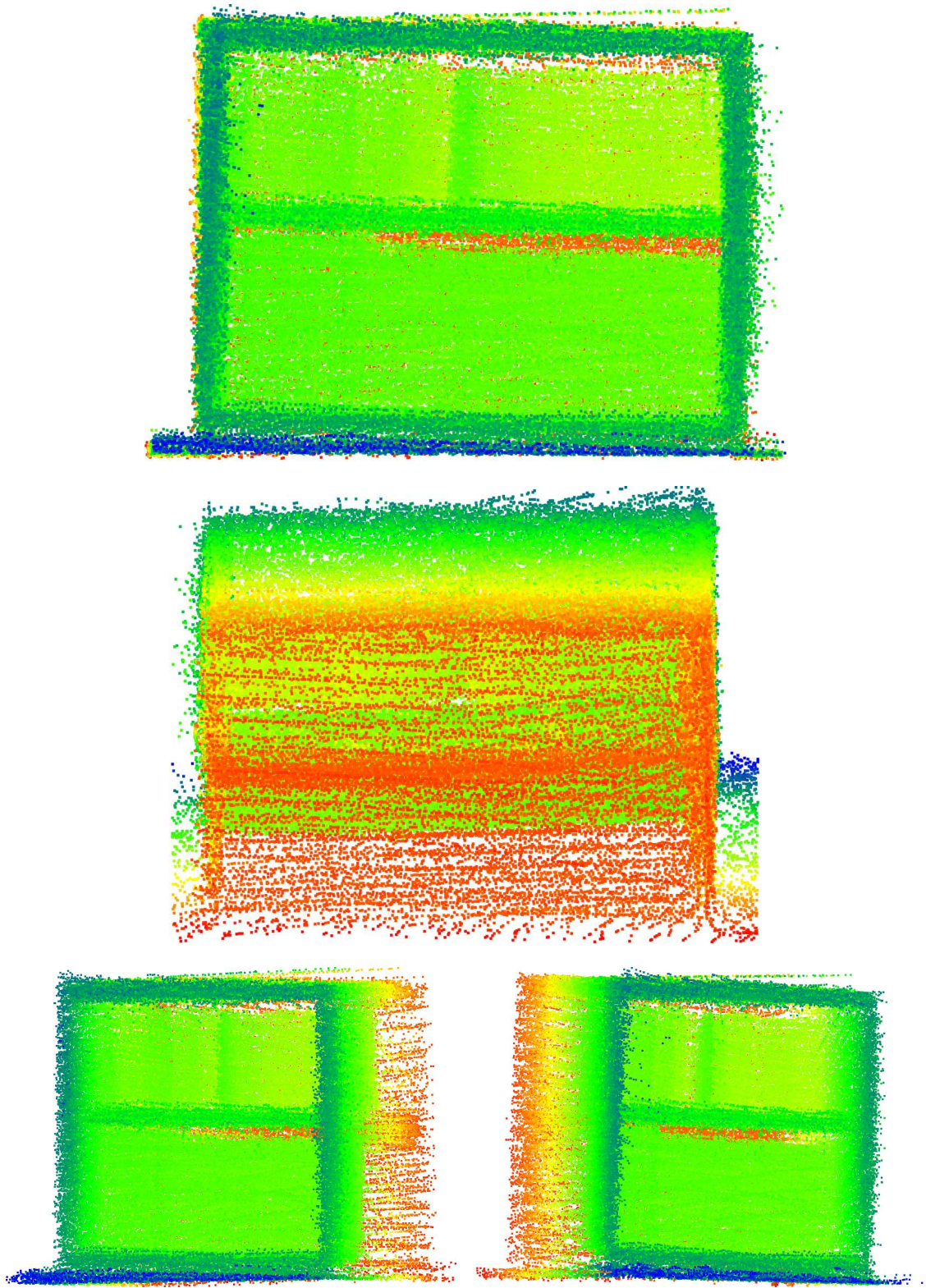


Figure A.20: The pointcloud result obtained from an observation of the small bookshelf using SEE++ with the Velodyne VLP-16. The images use a y -coordinate colourmap and show views of the front, back and two side angles of the bookshelf.



Figure A.21: Photographs of the rhinoceros pelvis scene. It contains the pelvis of a Javan rhinoceros (*rhinoceros sondaicus*) specimen that was loaned from the Oxford University Museum of Natural History (OUMNH 19164). The scene exhibits detailed visual features and surface texture with relatively complex geometry.

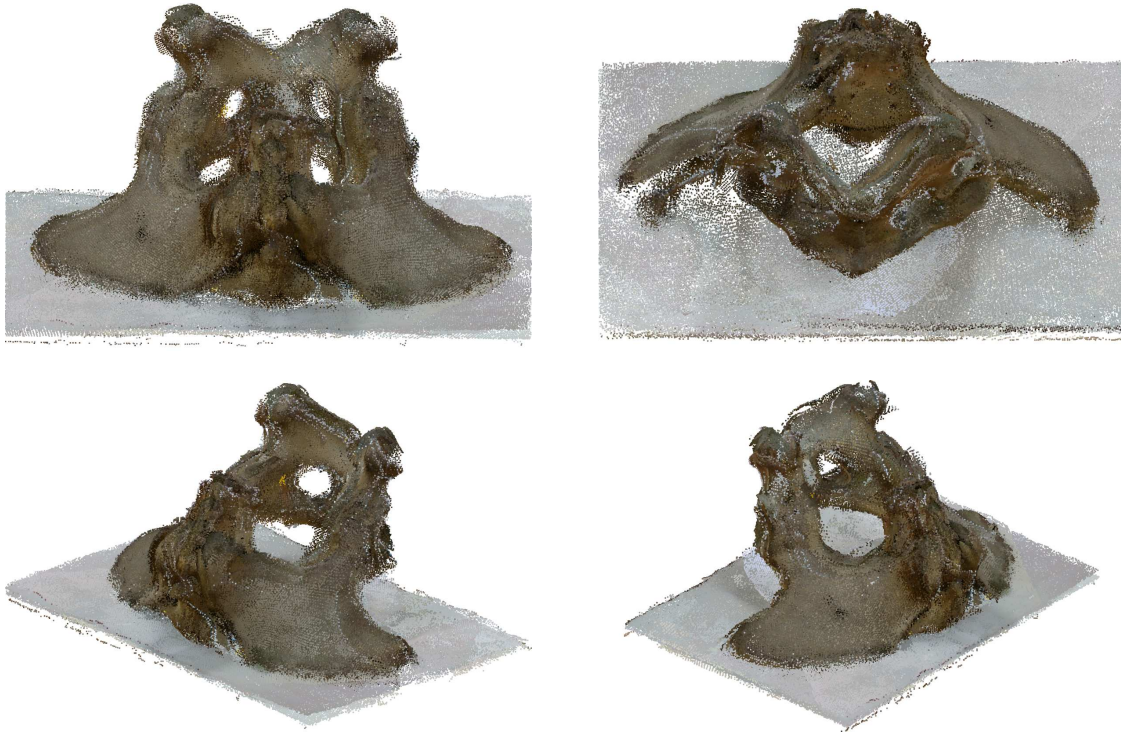


Figure A.22: The RGB coloured pointcloud result obtained from an observation of the rhinoceros pelvis scene using SEE with the Intel RealSense D435.

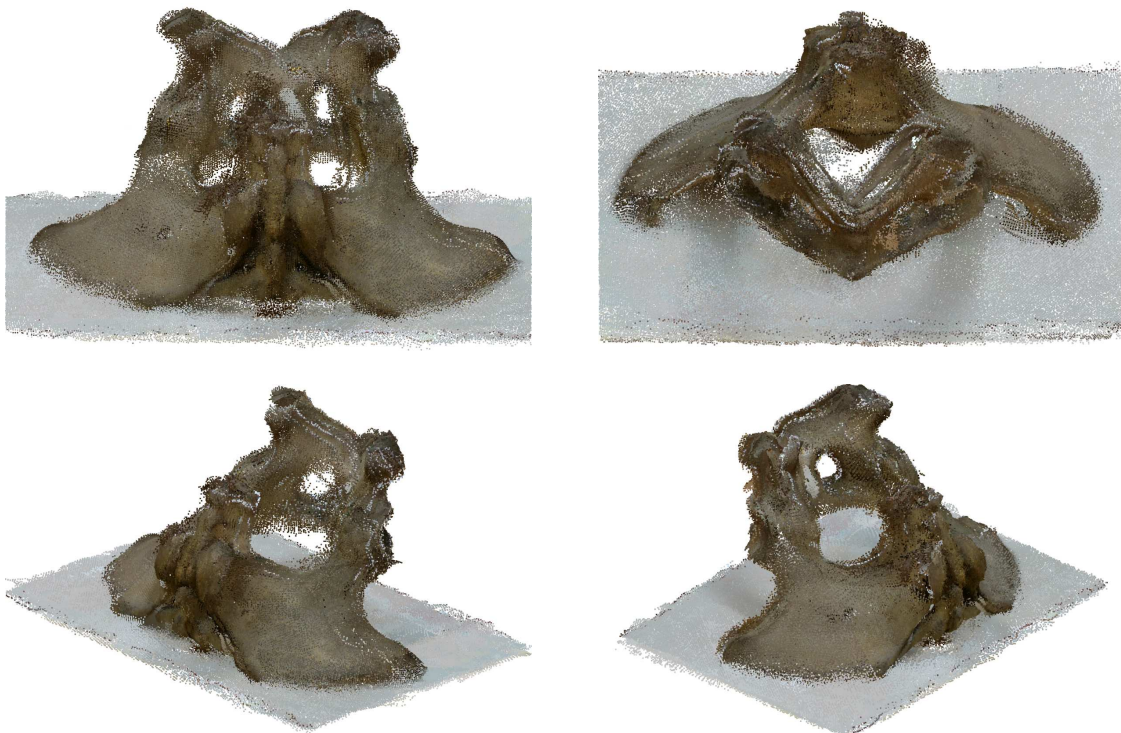


Figure A.23: The RGB coloured pointcloud result obtained from an observation of the rhinoceros pelvis scene using SEE++ with the Intel RealSense D435.

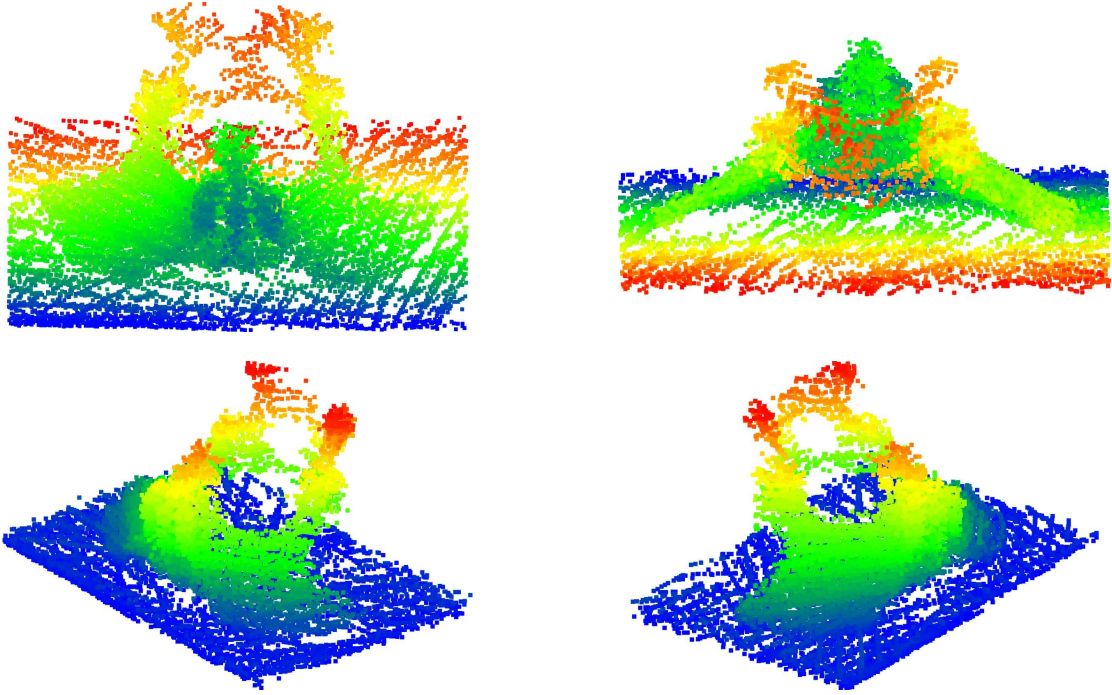


Figure A.24: The pointcloud result obtained from an observation of the rhinoceros pelvis scene using SEE with the Velodyne VLP-16. The top row of images use a y -coordinate colourmap and the bottom row of images use a z -coordinate colourmap.

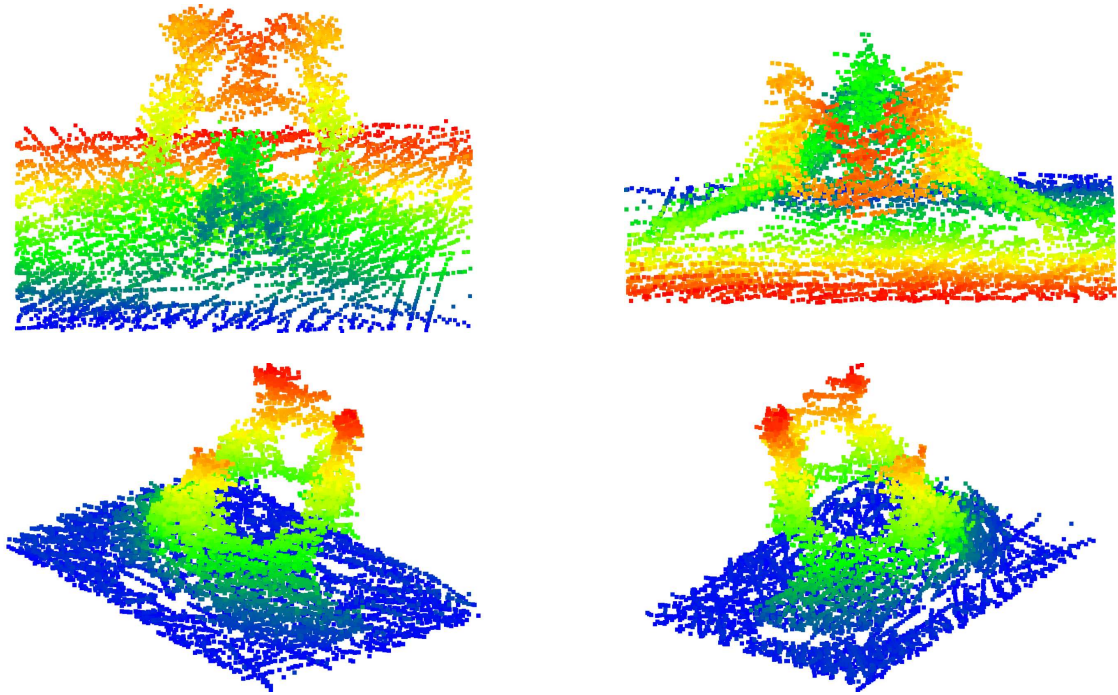


Figure A.25: The pointcloud result obtained from an observation of the rhinoceros pelvis using SEE++ with the Velodyne VLP-16. The top row of images use a y -coordinate colourmap and the bottom row of images use a z -coordinate colourmap.



Figure A.26: Photographs of the crocodile skull scene. It contains the skull of a saltwater crocodile (*Crocodylus porosus*) specimen that was loaned from the Oxford University Museum of Natural History (OUMNH 19149). The jaw was propped open using a tripod to enable the observation of interior surfaces.

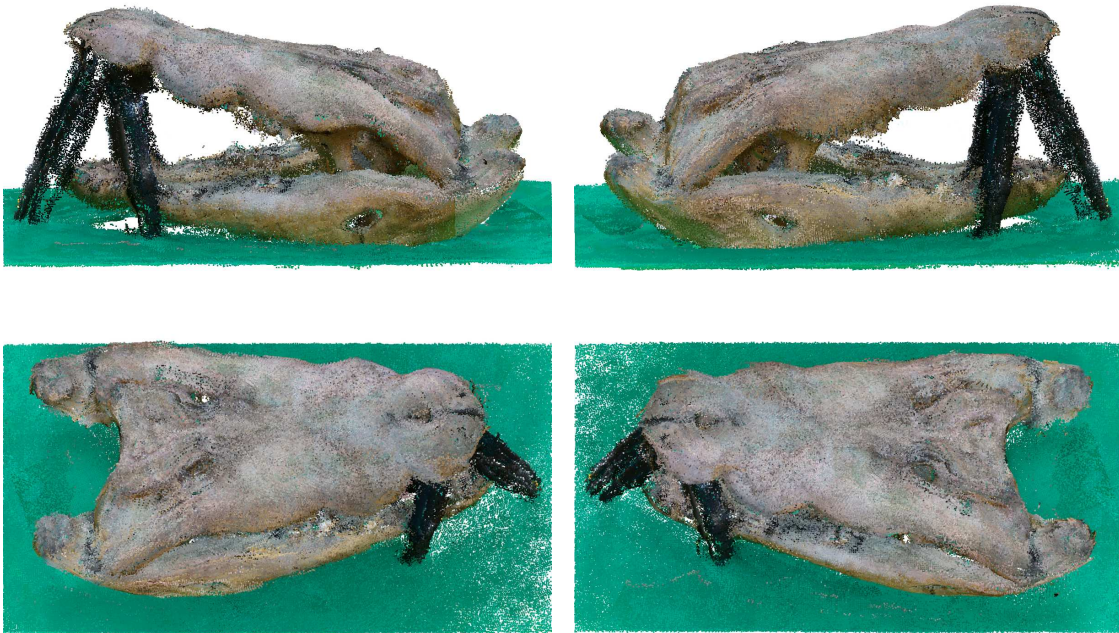


Figure A.27: The RGB coloured pointcloud result obtained from an observation of the crocodile skull scene using SEE with the Intel RealSense D435.

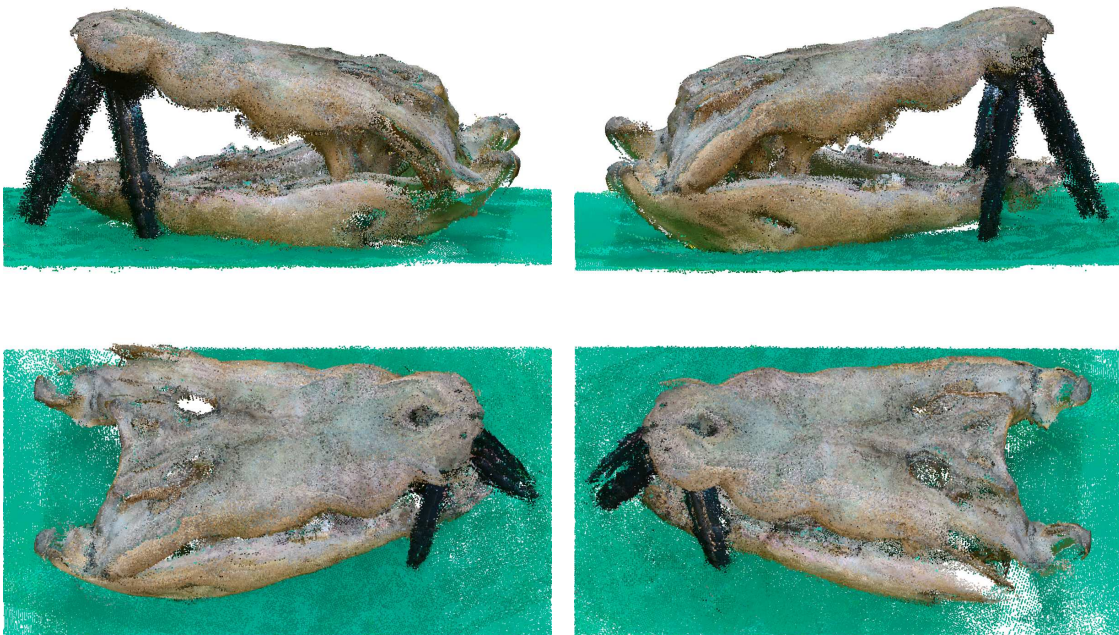


Figure A.28: The RGB coloured pointcloud result obtained from an observation of the crocodile skull scene using SEE++ with the Intel RealSense D435.

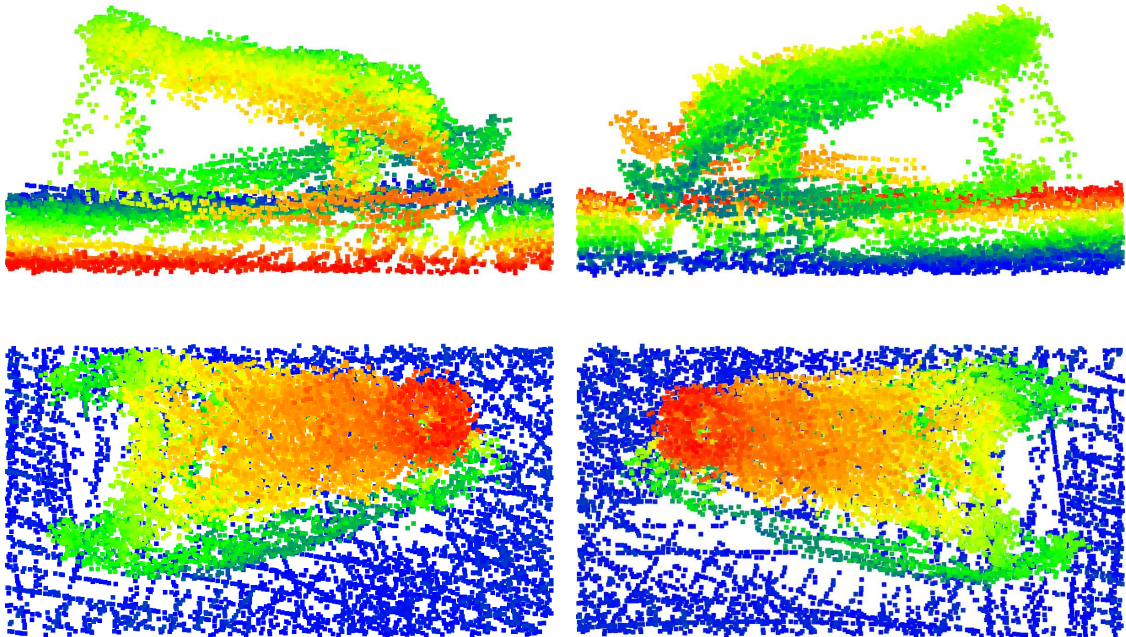


Figure A.29: The pointcloud result obtained from an observation of the crocodile skull scene using SEE with the Velodyne VLP-16. The top row of images use a y -coordinate colourmap and the bottom row of images use a z -coordinate colourmap.

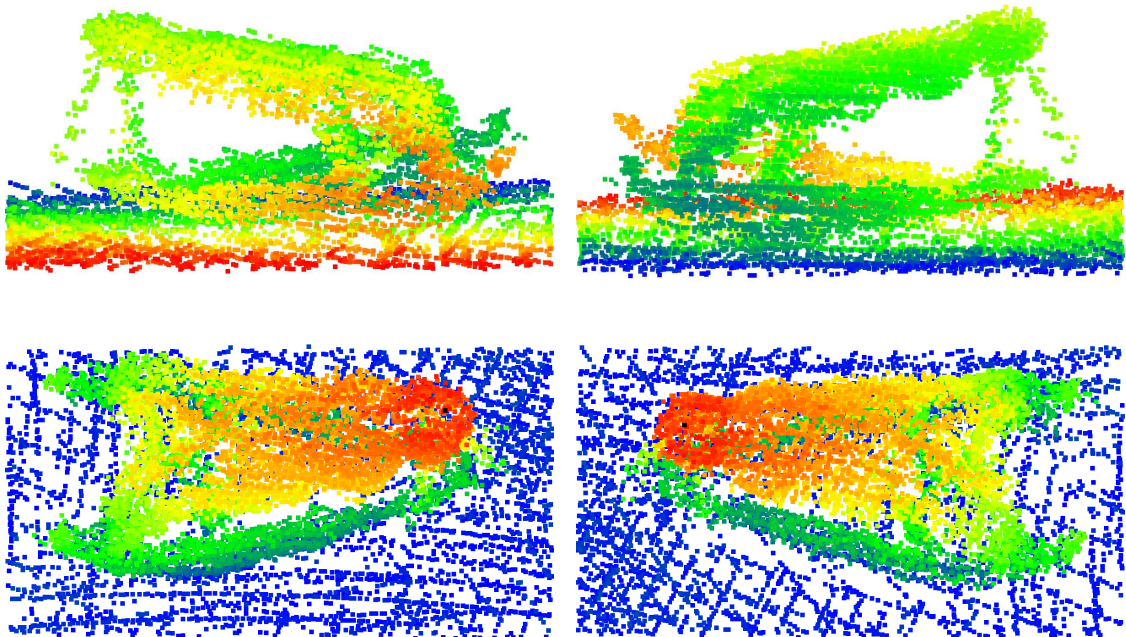


Figure A.30: The pointcloud result obtained from an observation of the crocodile skull using SEE++ with the Velodyne VLP-16. The top row of images use a y -coordinate colourmap and the bottom row of images use a z -coordinate colourmap.

References

- Abduldayem, Abdullah, Dongming Gan, Lakmal Seneviratne, and Tarek Taha (2017). “3D Reconstruction of Complex Structures with Online Profiling and Adaptive Viewpoint Sampling”. In: *International Micro Air Vehicle Conference and Flight Competition*, pp. 278–285.
- Adler, Benjamin, Junhao Xiao, and Jianwei Zhang (2013). “Finding next best views for autonomous UAV mapping through GPU-accelerated particle simulation”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1056–1061.
- Adler, Benjamin, Junhao Xiao, and Jianwei Zhang (2014). “Autonomous Exploration of Urban Environments using Unmanned Aerial Vehicles”. In: *Journal of Field Robotics* 31.6, pp. 912–939.
- Banta, Joseph E., Laurana M. Wong, Christophe Dumont, and Mongi A. Abidi (2000). “A next-best-view system for autonomous 3-D object reconstruction”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 30.5, pp. 589–598.
- Besl, Paul J. and Neil D. McKay (1992). “A Method for Registration of 3-D Shapes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2, pp. 239–256.
- Bircher, Andreas, Kostas Alexis, Michael Burri, Philipp Oettershagen, Sammy Omari, Thomas Mantel, and Roland Siegwart (2015). “Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics”. In: *IEEE International Conference on Robotics and Automation*, pp. 6423–6430.
- Bircher, Andreas, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart (2016). “Receding Horizon Next-Best-View Planner for 3D Exploration”. In: *IEEE International Conference on Robotics and Automation*, pp. 1462–1468.
- Bircher, Andreas, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart (2018). “Receding horizon path planning for 3D exploration and surface inspection”. In: *Autonomous Robots* 42.2, pp. 291–306.
- Blaer, Paul S. and Peter K. Allen (2007). “Data acquisition and view planning for 3-D modeling tasks”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 417–422.
- Bodenmüller, Tim (2009). “Streaming surface reconstruction from real time 3D measurements.” PhD thesis.
- Border, Rowan and Jonathan D. Gammell (2020). “Proactive Estimation of Occlusions and Scene Coverage for Planning Next Best Views in an Unstructured Representation”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–8.
- Border, Rowan, Jonathan D. Gammell, and Paul Newman (2017). “Inferring Surface Geometry from Point Clouds for Next Best View Planning”. In: *Joint Industry and Robotics CDTs Symposium*, pp. 1–2.

- Border, Rowan, Jonathan D. Gammell, and Paul Newman (2018). “Surface Edge Explorer (SEE): Planning Next Best Views Directly from 3D Observations”. In: *IEEE International Conference on Robotics and Automation*, pp. 1–8.
- Boronczyk, Jakub (2016). *Radcliffe Camera*. <https://bit.ly/2UZnNkJ>.
- Chen, Shengyong and Youfu Li (2005). “Vision sensor planning for 3-D model acquisition”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 35.5, pp. 894–904.
- Chen, Shengyong, Youfu Li, and Ngai Ming Kwok (2011). “Active vision in robotic systems: A survey of recent developments”. In: *The International Journal of Robotics Research* 30.11, pp. 1343–1377.
- Chvatal, Vasek (1979). “A Greedy Heuristic for the Set-Covering Problem”. In: *Mathematics of Operations Research* 4.3, pp. 233–235.
- Connolly, Christopher Ian (1985). “The determination of next best views”. In: *IEEE International Conference on Robotics and Automation*, pp. 432–435.
- Cover, Hugh, Sanjiban Choudhury, Sebastian Scherer, and Sanjiv Singh (2013). “Sparse Tangential Network (SPARTAN): Motion planning for micro aerial vehicles”. In: *IEEE International Conference on Robotics and Automation*. IEEE, pp. 2820–2825.
- Curless, Brian and Marc Levoy (1996). “A Volumetric Method for Building Complex Models from Range Images”. In: *SIGGRAPH Computer Graphics and Interactive Techniques*, pp. 303–312.
- Daudelin, Jonathan and Mark Campbell (2017). “An Adaptable, Probabilistic, Next-Best View Algorithm for Reconstruction of Unknown 3-D Objects”. In: *IEEE Robotics and Automation Letters* 2.3, pp. 1540–1547.
- Delmerico, Jeffrey, Stefan Isler, Reza Sabzevari, and Davide Scaramuzza (2018). “A comparison of volumetric information gain metrics for active 3D object reconstruction”. In: *Autonomous Robots* 42.2, pp. 197–208.
- Dierenbach, Kai O., Martin Weinmann, and Boris Jutzi (2016). “Next-Best-View method based on consecutive evaluation of topological relations”. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. 41, pp. 11–19.
- Drezner, Zvi and George O. Wesolowsky (1983). “Minimax and maximin facility location problems on a sphere”. In: *Naval Research Logistics* 30.2, pp. 305–312.
- Ester, Martin, Hans P Kriegel, Jorg Sander, and Xiaowei Xu (1996). “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *International Conference on Knowledge Discovery and Data Mining*, pp. 226–231.
- Fletcher, P. Thomas, Conglin Lu, Stephen M. Pizer, and Sarang Joshi (2004). “Principal geodesic analysis for the study of nonlinear statistics of shape”. In: *IEEE Transactions on Medical Imaging* 23.8, pp. 995–1005.
- Foissotte, Torea, Olivier Stasse, Adrien Escande, Pierre Brice Wieber, and Abderrahmane Kheddar (2009). “A two-steps next-best-view algorithm for autonomous 3D object modeling by a humanoid robot”. In: *International Conference on Robotics and Automation*, pp. 1159–1164.
- Fritzke, Bernd (1994). “A Growing Neural Gas Network Learns Topologies”. In: *International Conference on Neural Information Processing Systems*. Vol. 7, pp. 625–632.
- Hollinger, Geoffrey A., Brendan J. Englot, Franz S. Hover, Urbashi Mitra, and Gaurav S. Sukhatme (2012). “Active planning for underwater inspection and the benefit of adaptivity”. In: *International Journal of Robotics Research* 32.1, pp. 3–18.

- Hotelling, Harold (1933). "Analysis of a Complex of Statistical Variables Into Principal Components". In: *Journal of Educational Psychology* 24.6, pp. 417–441.
- Isler, Stefan, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza (2016). "An Information Gain Formulation for Active Volumetric 3D Reconstruction". In: *IEEE International Conference on Robotics and Automation*, pp. 3477–3484.
- Jung, Sungkyu, Ian L. Dryden, and J. S. Marron (2012). "Analysis of principal nested spheres". In: *Biometrika* 99.3, pp. 551–568.
- Kaba, Mustafa Devrim, Mustafa Gokhan Uzunbas, and Ser Nam Lim (2017). "A Reinforcement Learning Approach to the View Planning Problem". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5094–5102.
- Karaman, Sertac and Emilio Frazzoli (2011). "Sampling-based Algorithms for Optimal Motion Planning". In: *International Journal of Robotics Research* 30.7, pp. 846–894.
- Karaszewski, Maciej, Marcin Adamczyk, and Robert Sitnik (2016a). "Assessment of next-best-view algorithms performance with various 3D scanners and manipulator". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 119, pp. 320–333.
- Karaszewski, Maciej, Robert Sitnik, and Eryk Bunsch (2012). "On-line, collision-free positioning of a scanner during fully automated three-dimensional measurement of cultural heritage objects". In: *Robotics and Autonomous Systems* 60.9, pp. 1205–1219.
- Karaszewski, Maciej, Marcin Stepień, and Robert Sitnik (2016b). "Two-stage automated measurement process for high-resolution 3D digitization of unknown objects". In: *Applied Optics* 55.29, pp. 8162–8170.
- Karp, Richard (1972). "Reducibility Among Combinatorial Problems". In: *Complexity of Computer Computations*, pp. 85–103.
- Katz, Sagi, George Leifman, and Ayellet Tal (2005). "Mesh segmentation using feature point and core extraction". In: *Visual Computer* 21.8, pp. 649–658.
- Katz, Sagi, Ayellet Tal, and Ronen Basri (2007). "Direct visibility of point sets". In: *ACM Transactions on Graphics* 26.3, p. 24.
- Kavraki, Lydia E., Petr Švestka, Jean Claude Latombe, and Mark H. Overmars (1996). "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *IEEE Transactions on Robotics and Automation* 12.4, pp. 566–580.
- Kazhdan, Michael, Matthew Bolitho, and Hugues Hoppe (2006). "Poisson Surface Reconstruction". In: *Eurographics Symposium on Geometry Processing*, pp. 1–13.
- Khalfaoui, Souhail, Ralph Seulin, Yohan Fougerolle, and David Fofi (2013). "An efficient method for fully automatic 3D digitization of unknown objects". In: *Computers in Industry* 64.9, pp. 1152–1160.
- Kraft, Dieter (1994). "Algorithm 733: TOMP - Fortran modules for optimal control calculations". In: *ACM Transactions on Mathematical Software* 20.3, pp. 262–281.
- Kraft, Donald H. (1988). "A software package for sequential quadratic programming". In: *Technical Report DFVLR-FB*.
- Krainin, Michael, Brian Curless, and Dieter Fox (2011). "Autonomous generation of complete 3D object models using next best view manipulation planning". In: *IEEE International Conference on Robotics and Automation*, pp. 5031–5037.
- Kriegel, Simon, Tim Bodenmüller, Michael Suppa, and Gerd Hirzinger (2011). "A Surface-Based Next-Best-View Approach for Automated 3D Model Completion of Unknown Objects". In: *IEEE International Conference on Robotics and Automation*, pp. 4869–4874.
- Kriegel, Simon, Christian Rink, Tim Bodenmüller, Alexander Narr, Michael Suppa, and Gerd Hirzinger (2012). "Next-best-scan planning for autonomous 3D modeling". In:

- IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2850–2856.
- Kriegel, Simon, Christian Rink, Tim Bodenmüller, and Michael Suppa (2015). “Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects”. In: *Journal of Real-Time Image Processing* 10.4, pp. 611–631.
- Krishnamurthy, Venkat and Marc Levoy (1996). “Fitting smooth surfaces to dense polygon meshes”. In: *Conference on Computer Graphics and Interactive Techniques*, pp. 313–324.
- Kunze, Lars, Mohan Sridharan, Christos Dimitrakakis, and Jeremy Wyatt (2017). “View Planning with Time Constraints - An Adaptive Sampling Approach”. In: *IEEE International Conference on Robotics and Automation*, pp. 1–6.
- LaValle, Steven M. (1998). *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. Tech. rep.
- Levenberg, Kenneth (1943). “A Method for the Solution of Certain Non-Linear Problems in Least Squares”. In: *Quarterly of Applied Mathematics* 1.278, pp. 536–538.
- Low, Kok Lim and Anselmo Lastra (2006). “Efficient constraint evaluation algorithms for hierarchical next-best-view planning”. In: *Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 830–837.
- Marquardt, Donald W. (1963). “An Algorithm for Least-Squares Estimation of Nonlinear Parameters”. In: *Journal of the Society for Industrial and Applied Mathematics* 11.2, pp. 431–441.
- Massios, Nikolaos A. and Robert B. Fisher (1998). “A Best Next View Selection Algorithm Incorporating a Quality Criterion”. In: *British Machine Vision Conference*, pp. 780–789.
- McGreavy, Christopher, Lars Kunze, and Nick Hawes (2017). “Next best view planning for object recognition in mobile robotics”. In: *CEUR Workshop Proceedings* 1782.2.
- Mendoza, Miguel, Juan Irving Vasquez-Gomez, Hind Taud, Enrique Sucar, and Carolina Reta (2019). “Supervised Learning of the Next-Best-View for 3D Object Reconstruction”. In: *arXiv*, pp. 1–15.
- Monica, Riccardo and Jacopo Aleotti (2018a). “Contour-based next-best view planning from point cloud segmentation of unknown objects”. In: *Autonomous Robots* 42.2, pp. 443–458.
- Monica, Riccardo and Jacopo Aleotti (2018b). “Surfel-Based Next Best View Planning”. In: *IEEE Robotics and Automation Letters* 3.4, pp. 3324–3331.
- Newell, Martin Edward (1975). “The utilization of procedure models in digital image synthesis.” PhD thesis.
- OUMNH (19149). *Crocodylus Porosus*.
- OUMNH (19164). *Rhinoceros Sondaicus*.
- Papadopoulos-Orfanos, Dimitri and Francis Schmitt (1997). “Automatic 3-D digitization using a laser rangefinder with a small field of view”. In: *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pp. 60–67.
- Patel, Minnie H. and Arun Chidambaram (2002). “A new method for minimax location on a sphere”. In: *International Journal of Industrial Engineering: Theory Applications and Practice* 9.1, pp. 96–102.
- Pearson, Karl (1901). “On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, pp. 559–572.

- Peng, Cheng and Volkan Isler (2019). “Adaptive View Planning for Aerial 3D Reconstruction”. In: *IEEE International Conference on Robotics and Automation*, pp. 2981–2987.
- Pito, Richard (1996). “A sensor-based solution to the next best view problem”. In: *International Conference on Pattern Recognition* 1.10, pp. 941–945.
- Potthast, Christian and Gaurav S Sukhatme (2011). “Next best view estimation with eye in hand camera”. In: *International Conference on Intelligent Robots and Systems*, pp. 1–4.
- Potthast, Christian and Gaurav S. Sukhatme (2014). “A probabilistic framework for next best view estimation in a cluttered environment”. In: *Journal of Visual Communication and Image Representation* 25.1, pp. 148–164.
- Rasmussen, Carl Edward and Christopher K. I. Williams (2006). *Gaussian Processes for Machine Learning*.
- Reed, Michael K. and Peter K. Allen (2000). “Constraint-based sensor planning for scene modeling”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.12, pp. 1460–1467.
- Roberts, Mike, Anh Truong, Debadepta Dey, Sudipta Sinha, Ashish Kapoor, Neel Joshi, and Pat Hanrahan (2017). “Submodular Trajectory Optimization for Aerial 3D Scanning”. In: *International Conference on Computer Vision*, pp. 5334–5343.
- Rodrigues, Olinde (1840). “Des Lois Geometriques Qui Regissent les Deplacements d’un Systeme Solide dans L’espace, et de la Variation des Coordonnees Provenant de ces Deplacements Consideres dependamment des Causes Qui Peuvent les Produire”. In: *Journal de Mathématiques Pures et Appliquées* 5.1840, pp. 380–440.
- Scott, William R. (2009). “Model-based View Planning”. In: *Machine Vision and Applications* 20.1, pp. 47–69.
- Scott, William R., Gerhard Roth, and Jean François Rivest (2003). “View planning for automated three-dimensional object reconstruction and inspection”. In: *ACM Computing Surveys* 35.1, pp. 64–96.
- Selin, Magnus, Mattias Tiger, Daniel Duberg, Fredrik Heintz, and Patric Jensfelt (2019). “Efficient autonomous exploration planning of large-scale 3-d environments”. In: *IEEE Robotics and Automation Letters* 4.2, pp. 1699–1706.
- Song, Soohwan and Sungho Jo (2017). “Online inspection path planning for autonomous 3D modeling using a micro-aerial vehicle”. In: *IEEE International Conference on Robotics and Automation*, pp. 6217–6224.
- Song, Soohwan and Sungho Jo (2018). “Surface-Based Exploration for Autonomous 3D Modeling”. In: *IEEE International Conference on Robotics and Automation*, pp. 1–8.
- Tarabanis, Konstantinos A, Peter K Allen, and Roger Y Tsai (1995). “A Survey of Sensor Planning in Computer Vision”. In: *IEEE Transactions on Robotics and Automation* 11.1, pp. 86–104.
- Tarbox, Glenn H. and Susan N. Gottschlich (1995). “Planning for Complete Sensor Coverage in Inspection”. In: *Computer Vision and Image Understanding* 61.1, pp. 84–111.
- Trummer, Michael, Christoph Munkelt, and Joachim Denzler (2010). “Online next-best-view planning for accuracy optimization using an extended E-criterion”. In: *International Conference on Pattern Recognition*, pp. 1642–1645.
- Turk, Greg and Marc Levoy (1994). “Zippered polygon meshes from range images”. In: *SIGGRAPH Computer Graphics and Interactive Techniques*, pp. 311–318.

- Vasquez-Gomez, Juan Irving, Efraín Lopez-Damian, and Luis Enrique Sucar (2009). “View planning for 3D object reconstruction”. In: *International Conference on Intelligent Robots and Systems*, pp. 4015–4020.
- Vasquez-Gomez, Juan Irving, Luis Enrique Sucar, and Rafael Murrieta-Cid (2013). “Hierarchical ray tracing for fast volumetric next-best-view planning”. In: *International Conference on Computer and Robot Vision*, pp. 181–187.
- Vasquez-Gomez, Juan Irving, Luis Enrique Sucar, and Rafael Murrieta-Cid (2017). “View/state planning for three-dimensional object reconstruction under uncertainty”. In: *Autonomous Robots* 41.1, pp. 89–109.
- Vasquez-Gomez, Juan Irving, Luis Enrique Sucar, Rafael Murrieta-Cid, and Juan Carlos Herrera-Lozada (2018). “Tree-based search of the next best view/state for three-dimensional object reconstruction”. In: *International Journal of Advanced Robotic Systems* 15.1, pp. 1–11.
- Vasquez-Gomez, Juan Irving, Luis Enrique Sucar, Rafael Murrieta-Cid, and Efraín Lopez-Damian (2014). “Volumetric Next Best View Planning for 3D Object Reconstruction with Positioning Error”. In: *International Journal of Advanced Robotic Systems* 11.10, p. 159.
- Wenhardt, Stefan, Benjamin Deutsch, Elli Angelopoulou, and Heinrich Niemann (2007). “Active visual object reconstruction using D-, E-, and T-optimal next best views”. In: *Conference on Computer Vision and Pattern Recognition*, pp. 1–7.
- Wong, Laurana M., Christophe Dumont, and Mongi A. Abidi (1999). “Next Best View System in a 3-D Object Modeling Task”. In: *International Symposium on Computational Intelligence in Robotics and Automation*, pp. 306–311.
- Wu, Zhirong, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao (2015). “3D ShapeNets: A deep representation for volumetric shapes”. In: *Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920.
- Yang, Chen and Gérard Medioni (1992). “Object modelling by registration of multiple range images”. In: *Image and Vision Computing* 10.3, pp. 145–155.
- Yoder, Luke and Sebastian Scherer (2016). “Autonomous Exploration for Infrastructure Modeling with a Micro Aerial Vehicle”. In: *Field and Service Robotics* 10, pp. 427–440.
- Yuan, Xiaobu (1995). “A mechanism of automatic 3D object modeling”. In: *Transactions on Pattern Analysis and Machine Intelligence* 17.3, pp. 307–311.