Data Driven Selective Sensing for 3D Image Acquisition

By:

Phillip Curtis

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements for the

Doctorate in Philosophy degree in Electrical and Computer Engineering

Ottawa-Carleton Institute of Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

It is well established that acquiring large amounts of range data with vision sensors can quickly lead to important data management challenges where processing capabilities become saturated and pre-empt full usage of the information available for autonomous systems to make educated decisions. While sub-sampling offers a naïve solution for reducing dataset dimension after acquisition, it does not capitalize on the knowledge available in already acquired data to selectively and dynamically drive the acquisition process over the most significant regions in a scene, the latter being generally characterized by variations in depth and surface shape in the context of 3D imaging.

This thesis discusses the development of two formal improvement measures, the first based upon surface meshes and Ordinary Kriging that focuses on improving scene accuracy, and the second based upon probabilistic occupancy grids that focuses on improving scene coverage. Furthermore, three selection processes to automatically choose which locations within the field of view of a range sensor to acquire next are proposed based upon the two formal improvement measures. The first two selection processes each use only one of the proposed improvement measures. The third selection process combines both improvement measures in order to counterbalance the parameters of the accuracy of knowledge about the scene and the coverage of the scene.

The proposed algorithms mainly target applications using random access range sensors, defined as sensors that can acquire depth measurements at a specified location within their field of view. Additionally, the algorithms are applicable to the case of estimating the improvement and point selection from within a single point of view, with the purpose of guiding the random access sensor to locations it can acquire. However, the framework is developed to be independent of the range sensing technology used, and is validated with range data of several scenes acquired from many different sensors employing various sensing technologies and configurations. Furthermore, the experimental results of the proposed selection processes are compared against those produced by a random sampling process, as well as a neural gas selective sensing algorithm.

Acknowledgments

I would like to thank my supervisor, Dr. Pierre Payeur, for his support and funding over the course of my research. Secondly, I would like to thank the University of Ottawa for providing me the opportunity and funding to obtain a Ph.D. Thirdly, I would like to thank NSERC, who provided me with a scholarship for the first few years of study. Finally, I would like to give my thanks to my family and friends, for providing the emotional support and encouragement, without which I would not have been able to succeed.

Table of Contents

List of Figures

List of Tables

Acronyms

Acronym	Meaning
ASLS	Adaptive Structured Light Sensor
ICP	Iterative Closest Point
IR	Infra Red
LS	Least Square
MDP	Markov Decision Process
MSE	Mean Squared Error
NBV	Next Best View
OPDF	Occupancy Probability Distribution Function
PCA	Principal Component Analysis
POG	Probabilistic Occupancy Grid
POMDP	Partially Observable Markov Decision Process
PSNR	Peak Signal to Noise Ratio
RBF	Radial Basis Function
RGB	Red Green Blue
SM	Surface Mesh
VMR	Variance to Mean Ratio

List of Symbols

Symbol	Meaning
$\boldsymbol{p_i}$	i^{th} known (currently measured) data point
$\widehat{\boldsymbol{p}}_{\boldsymbol{j}}$	j^{th} unknown (currently unmeasured) data point
$\boldsymbol{n}$	The number of known, or measured, data points
$\boldsymbol{m}$	The number of unknown, or currently unmeasured, data points
$\boldsymbol{h}(\boldsymbol{p_i}, \boldsymbol{p_j})$	The distance between points $\boldsymbol{p_i}$ and $\boldsymbol{p_j}$, such that if $\boldsymbol{p_i} = (\boldsymbol{c_{i0}}, \boldsymbol{c_{i1}})$ and if $\boldsymbol{p_j} = (\boldsymbol{c_{j0}}, \boldsymbol{c_{j1}})$, then $\boldsymbol{h}(\boldsymbol{p_i}, \boldsymbol{p_j}) = \sqrt{(\boldsymbol{c_{j0}} - \boldsymbol{c_{i0}})^2 + (\boldsymbol{c_{j1}} - \boldsymbol{c_{i1}})^2}$
$\boldsymbol{\gamma}\left(\boldsymbol{h}(\boldsymbol{p_i}, \boldsymbol{p_j})\right)$	The semivariogram for the value of the distance measure between points $\boldsymbol{p_i}$ and $\boldsymbol{p_j}$
$\boldsymbol{\gamma_{emp}}\left(\boldsymbol{h}(\boldsymbol{p_i}, \boldsymbol{p_j})\right)$	The empirical semivariogram for the value of the distance measure between points $\boldsymbol{p_i}$ and $\boldsymbol{p_j}$
$\boldsymbol{K}$	The data to data covariance matrix - an $(n+1) \times (n+1)$ matrix
$\widehat{\boldsymbol{K}}(\widehat{\boldsymbol{p}}_{\boldsymbol{s}})$	The data to data covariance matrix, if we assume that $\widehat{\boldsymbol{p}}_s$ is the $(n+1)^{th}$ known data point - an $(n+2) \times (n+2)$ matrix
$\boldsymbol{k}(\widehat{\boldsymbol{p}}_{\boldsymbol{j}})$	The data to unknown covariance vector, where the semivariogram between the n known data points is calculated with respect to the j^{th} unknown data location - an $(n+1)$ sized vector
$\widehat{\boldsymbol{k}}(\widehat{\boldsymbol{p}}_{\boldsymbol{j}}\|\widehat{\boldsymbol{p}}_{\boldsymbol{s}})$	The data to unknown covariance vector, if we assume that $\widehat{\boldsymbol{p}}_s$ is the $(n+1)^{th}$ known data point - an $(n+2)$ sized vector
$\boldsymbol{z}(\boldsymbol{p_i})$	The value at the i^{th} known data point
$\boldsymbol{Z}$	The known data value vector for each of the n known data points - an $(n+1)$ sized vector
$\hat{\boldsymbol{z}}(\widehat{\boldsymbol{p}}_{\boldsymbol{j}})$	The estimated data value at the j^{th} unknown data point
$\widehat{\boldsymbol{Z}}(\widehat{\boldsymbol{p}}_{\boldsymbol{s}})$	The known data value vector, if we assume that $\widehat{\boldsymbol{p}}_s$ is the $(n+1)^{th}$ known data point - an $(n+2)$ sized vector
$\hat{\boldsymbol{z}}(\widehat{\boldsymbol{p}}_{\boldsymbol{j}}\|\widehat{\boldsymbol{p}}_{\boldsymbol{s}})$	The estimated data value at the j^{th} unknown data point, if we assume that $\widehat{\boldsymbol{p}}_s$ is the $(n+1)^{th}$ known data point

Symbol	Meaning
$\boldsymbol{\lambda(\hat{p}_j)}$	The Kriging weight vector corresponding to the j^{th} unknown data point - an $(n+1)$ sized vector
$\boldsymbol{\lambda(\hat{p}_j \mid \hat{p}_s)}$	The Kriging weight vector corresponding to the j^{th} unknown data point, if we assume that $\hat{\boldsymbol{p}}_s$ is the $(n+1)^{th}$ known data point - an $(n+2)$ sized vector
$\boldsymbol{\hat{\sigma}^2(\hat{p}_j)}$	The estimated Kriging variance corresponding to the estimated value at the j^{th} unknown data point
$\boldsymbol{\hat{\sigma}^2(\hat{p}_j \mid \hat{p}_s)}$	The estimated Kriging variance corresponding to the estimated value at the j^{th} unknown data point, if we assume that $\hat{\boldsymbol{p}}_s$ is the $(n+1)^{th}$ known data point
$\boldsymbol{imp_{SM}(\hat{p}_j \mid \hat{p}_s)}$	The estimated improvement for the estimation of the value at unknown location $\hat{\boldsymbol{p}}_j$ if we assume that the next point we acquire is at $\hat{\boldsymbol{p}}_s$ and that there will be zero error between $\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}_s)$ and $\boldsymbol{z}(\hat{\boldsymbol{p}}_s)$
$\boldsymbol{imp_{SM}(\hat{p}_s)}$	The total estimated improvement if we assume that the next point we acquire is at $\hat{\boldsymbol{p}}_s$ and that there will be zero error between $\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}_s)$ and $\boldsymbol{z}(\hat{\boldsymbol{p}}_s)$

Chapter 1. Introduction

This thesis develops methods to perform selective sensing within a static single point of view, with the goal of providing this capability at the time of acquisition. Within the context of this thesis, selective sensing is defined as intelligently selecting where to next acquire range data, based upon previously acquired knowledge of the same scene, whether it be an initial coarse acquisition of data, or a progressively built up data model. Single point of view means that the integrated sensor apparatus will not be moved such that non-naïve data fusion techniques would be required. This thesis targets slow random access range sensors, but is general enough to be applied to any range sensor. The first section of this chapter discusses the motivation for this work, the second discusses the objectives of the thesis, and the third section discusses the structure of the document.

1.1 Motivation and Context

Intelligent data acquisition and filtering will play an important part in sensing and machine vision in the future. With the current rate of progression in semiconductor technologies following the exponential rate stated in Moore's law [1], which is the observation that transistor counts and densities double every two years, the density and precision of sensors is increasing while using less material and operational power. With this trend, sensor costs are declining, and as a result an increasing number of sensors are being deployed on robotic platforms. In fact, data is being produced at a much faster rate than Moore's law [2], such that it is a challenge coping with additional processing demands, for both human and machine, imposed by complex scenarios such as military applications, surveillance systems, and autonomous operations [3], especially when real-time operations are desired. Additionally, the transistors themselves are becoming faster, which is directly leading to an increase in the sampling rates of sensors. Unfortunately, the drawback to this technological progression is that the time it takes to process and interpret the results from the more abundant data also increases, effectively lowering the amount of time to make a decision based upon the data for urgent scenarios. For example, by increasing the resolution of an imaging device by a factor of two in both the horizontal and vertical direction, four times the data is produced. This means that, assuming that the processing algorithm is linear with respect to the data size in performance, there will be a four times increase in the execution

time, but any processing speed increases from faster transistors is quickly cancelled out by the faster means to acquire datasets, and the increasing plurality of sensors. Furthermore, with the additional datasets, the attention of the end viewer, whether it be man or machine, is divided between the various representations of the many datasets, leading to information overload, which results in lower productivity in humans, and longer processing time in machines. The efficient handling of this convergence of many sources of data, with an increasing quantity of data, is known as the 'Big Data' challenge [3,4].

There are many possible ways to help with the 'Big Data' challenge, such as determining and displaying only the important information for the current task, or performing rich data fusion techniques for enhancing the displayed data to aid in the simplification of the consumption of the datasets by the end user. In the scope of this thesis, the transmission and integration of the acquired data into a larger database is not directly considered, but instead this thesis fixates on the minimization of the data collection for the specialized field of range image acquisition. This thesis relies on the assumption that in a range image acquisition, many of the acquired data points are often not really of interest. This is due the fact that a depth measurement made at a particular location is going to share certain characteristics in common with its neighbours if it belongs to the same object, and hence they may be predicted to a high confidence level from other points in the local neighbourhood. Regions that contain a lower level of confidence indicate areas that should be scanned in higher detail, such as where discontinuities lie or where there is sparse coverage. Based on this assumption, fewer points are needed to be acquired in order to get a similar resulting quality of representation of the scene compared to brute force high resolution acquisitions. Additionally, since not all range sensors are like common RGB camera images where all the raw data can be acquired virtually simultaneously, at least from the point of the user, with a single shot acquisition, these highly similar points each have a time to acquire cost in addition to the computational costs involved in future data processing stages. By reducing the data acquisition, in both time and data complexity but without significantly compromising on the amount of knowledge available, the end goal is to provide more time for advanced functionalities and decisions to be made, with the added benefit of minimizing transmission bandwidth. This becomes a paramount concern in remote robotic exploration tasks, such as in the case of the recent Martian rovers, as there is a limit to onboard

resources, as well as communication bandwidth [5]. This reduction of the number of points acquired without loss of significant details can be accomplished through selective sensing, based upon the currently available knowledge of that same scene from the same point of view.

1.2 Objectives

This thesis produces contributions to the field of intelligent sensing by providing techniques for efficiently acquiring a scene that can be used in critical applications such as autonomous robotics, search and rescue operations, surveillance systems, and remote systems monitoring. The main objectives of this research that result in original contributions are as follows:

1) Develop formal methods to quantitatively predict the improvement that acquiring specific sets of 3D points/regions would provide, based upon the current knowledge of a scene.
2) Develop formal methods to automatically and efficiently select the optimal location for the next acquisition to optimize the total potential improvement that future acquisitions can provide based upon the estimated improvement metrics previously developed, leading to a progressive 3D modelling framework.
3) Implement, adapt, and experimentally evaluate the proposed methods on data acquired from a variety of real range sensors, and demonstrate their independence to the context of the scene.

To further elaborate on contribution 1), which is the development of the improvement metrics, the surface mesh improvement measure developed in Section 3.1 takes into account accuracy, while the probabilistic occupancy grid improvement measure developed in Section 3.2 takes into account coverage. The development of the improvement metrics has the additional focus of ensuring that they are computationally feasible in order to support fast 3D scanning, while at the same time taking into account existing representation schemes such that data reuse can occur for future processing stages (which are beyond the scope of this thesis). This is accomplished by making appropriate approximations where necessary, in addition to using probabilistic occupancy grids and surface meshes representations, which are used extensively in 3D imaging and robotics applications. Furthermore, the metrics, and the models they are based on, are developed such that they can be incrementally added to

during acquisition, thereby providing a mechanism for information feedback to the selective sensing processes.

On contribution 2), which is the development of selection processes using the improvement metrics developed in contribution 1), three different selection processes are developed. The first selection process is guided by the surface mesh improvement measure (focuses solely on accuracy), and is detailed in Section 3.3.1. The second selection process is guided by the probabilistic occupancy grid improvement measure (focuses solely on coverage), and is detailed in Section 3.3.2. The third, and final, selection process is guided by and combines both improvement measures in order to drive selective acquisition by a balance of coverage and accuracy in an integrated manner. This topic is detailed in Section 3.3.3. Furthermore, the overall goal of these point selection processes is to choose a location that optimizes the overall potential improvements that future acquisitions can provide to the model of the scene. These point selection processes are developed so that changes in the underlying improvement map due to new point acquisitions are monitored, thereby allowing for efficient feedback on locations that should be acquired next in an iterative and incremental fashion.

On contribution 3), which is the implementation and experimental validation of the improvement metrics and the three selection processes made in contributions 1) and 2), several evaluations are made. The surface mesh improvement measure and its corresponding selection process are evaluated extensively using data obtained from a Microsoft Kinect sensor in Section 5.2. The probabilistic occupancy grid improvement measure and its corresponding selection process are evaluated also using data obtained from a Microsoft Kinect sensor in Section 5.3. The final combined selection process using both improvement measures is evaluated in Section 5.4, against the surface mesh improvement based selection process (the first selection processes proposed), the probabilistic occupancy grid improvement based selection process (the second selection process proposed), and against random point selection, using many sensors including a Microsoft Kinect, a Jupiter laser line scanner, a SICK Laser Metrology Scanner (LMS), an in-house adaptive structured light sensor (ASLS), and a Neptec LMS over a variety of scenes exhibiting several characteristics. By applying the proposed algorithms on datasets acquired from different sensors employing

diverse acquisition technologies, the independence of the proposed techniques to sensing methodology is demonstrated.

1.3 Document Structure

In this thesis, firstly an extensive review of the current literature is presented in Chapter 2, containing a review of current selective sensing methodologies, as well as reviews of techniques for acquiring, manipulating, and presenting 3D data. Additionally, the inclusion of a review on algorithms and techniques used in making decisions is provided. Secondly, Chapter 3 details the development of the different core algorithms proposed for the intelligent selective sensing methods presented in this thesis. This includes the development of an original surface mesh based measure of improvement which ensures that the estimated 3D model representation of the scene is accurately detailed, an original probabilistic occupancy grid measure of improvement which ensures that estimated 3D model representation covers all objects and features within the field of view of the sensor, and finally the original selection processes used for determining where to acquire future points based upon the previously developed improvement measures. Chapter 4 provides details of the experimental setup used for the evaluation of the developed techniques. Chapter 5 presents the application of the developed algorithms on several different scenes, which are experimentally acquired with various sensing technologies, in order to demonstrate their effectiveness, their independence to the sensing technology, and to the scene's content. An analysis is provided that uses random point selection, as well as a neural gas network selective sensing strategy, as a basis of comparison using several metrics. Chapter 6 presents the concluding remarks, states the original contributions of the work, and details future work and potential applications.

Chapter 2. Literature Review

In this chapter, a review of various techniques for the acquisition, estimation, handling, and representation of range images is made. Firstly, a discussion of the intelligent guiding of data acquisition is made that provides an overview of the current state-of-the-art in intelligent selective sensing. Secondly a review of estimation techniques is presented, as an intelligent selective sensing algorithm has need of estimating values, errors, and other properties such that it can properly select from where to acquire the next range measurement. Thirdly, decision processes are discussed, since in an autonomous application, an intelligent means of selecting the next acquisition point is required, as it is impractical to have human operator intervention at such a low level when several decisions have to be made in less than a second. Fourthly, techniques for range data representation, and methods for manipulating range datasets are presented.

2.1 Intelligent Data Acquisition

Intelligent data acquisition uses the currently known information about an object or a scene to guide future collection efforts in order to optimize the quantity of data gathered based on some predefined set of parameters. This thesis deals with the problem of intelligently selecting a location to measure the range from within a single point of view of a sensor. There are not many techniques that explicitly handle this single point of view problem. However, the multiple viewpoint intelligent data acquisition problem that deals with selecting the next viewpoint from which to acquire a ranged dataset of the object or environment has been investigated in many works. The multiple viewpoint problem provides some guidance to solving the single point of view problem, and it can be considered that the single point of view problem is a subset of the multiple view point problem. This section will discuss the current techniques for intelligent sensing within a single point of view, as well as a class of techniques which deal with intelligent sensing from multiple viewpoints, known as 'Next Best View' (NBV) algorithms, and finally provides a concluding analysis of their benefits and limitations.

2.1.1 Single Point of View

This section reviews different techniques which determine optimal methods for acquiring a range image from a single point of view. Some of the methods presented are adaptive in

nature, and can be considered as approaches to intelligent sensing, while others evaluate and discuss static scanning patterns which optimize certain criteria that are important to specific applications.

Cretu *et al.* [6,7] use a neural gas network approach for determining which regions of a 3D image need further scanning. Their technique first involves a coarse low resolution acquisition of the scene. The acquired point cloud is then fed into a neural gas network, which clusters points of similar distributions and features together. This clustering occurs as a result of the neural gas networks self-organizing principle that causes each individual neuron of the network being attracted to a region of points exhibiting similar statistical characteristics. As the networked is trained, the neural weights are adjusted such that each neuron, and the clusters of points that they effectively represent are expanded or contracted as appropriate. In the typical neural gas network, neurons are greedy, which ensures that the clusters and their representative neurons are as uniformly distributed as possible. In this application, however, training of the neural network is stopped early to ensure that the resulting clusters of points are not uniformly distributed, but instead are positioned in regions of high variability. The neural gas clusters are connected using Delaunay triangulation. Clusters that are closer together indicate that there is more variability along that region of the surface, as each cluster represents a different region consisting of different characteristics. As a result of the clusters being connected by the edges of triangles, triangles with edges shorter than the average define the regions with higher variability that are selected to be rescanned at a higher resolution. The drawback to the method is that the training of the neural network takes a large amount of time, and is highly reliant on the neural gas network from being overly trained, but it does provide coverage of the scene in regards to regions that need to be acquired with higher accuracy.

Shih *et al.* [8] develop three different techniques to guide a non-uniform data acquisition based on an unknown scene. In the first approach, an initial scan of three or four points of the region is acquired, such that the object is bounded by the enclosed geometry (a triangle if using three points, and a rectangle if using four points). Depending on the shape used, three or four extra points are acquired, creating four more sub-shapes, which are stored in a tree structure. Using plane fitting at each level of the tree, an error can be calculated between

actual triangles and estimated values between the current tree depth and previous tree depth. If the error is large enough, another subdivision, and hence acquisition along the current branch of the tree, is required. The second method uses curve fitting to generate a model based on an initial acquisition. After determining the inflection point of these curves, an approximation of the error over the intervals between inflection points is determined. If the error along each particular interval is large enough, a sampling of the mid-point of the interval is made, and the process repeats. The third and final method relies on knowing the CAD model of the object of which the scan is being made. An optimal scanning pattern of N points can be determined by a local adjustment technique, by first performing a simulated scan of the CAD model. A triangular mesh is applied to the scan, and then for each point in the mesh, an encompassing polygon is determined. A new point location is then determined by minimizing the mean squared errors of the triangular patches making up the polygon. This process is repeated until the points cease moving or another limit is reached. The first technique works well for planar objects, but does not improve upon brute force data acquisition for objects with large amount of curves. The second approach works well for curved objects as well as planar objects. The final method relies on a controlled environment and knowledge of the object being acquired and as such does not work well in an uncontrolled/unknown environment.

Ho and Saripalli [9] approach the problem of sensing strategy, based upon the energy function in the context of autonomous underwater vehicles (AUV's). An AUV has one primary constraint when trying to acquire information about its environment using its sensors: the stored energy capacity available for movement. This means that there is a direct trade-off between an accurate representation of its environment and how far it can travel. The authors investigate several different sampling strategies to determine an optimal scanning pattern under the constraints of improving accuracy and energy consumption for locomotion. Their results yield that a regular spiral pattern produces the least energy use and highest accuracy when compared to a random scanning pattern and a ‘lawnmower’ scanning pattern where the acquisition happens left to right on one row, then right to left on the next row.

Mandow *et al.* [10] develop a method to ensure a uniform distribution in space by performing subsampling of data acquired from a SICK LMS laser line scanner mounted on a tilt device. The primary motivation behind the work, is that the acquisition methodology inherently provides a varying degree of sampling density across the sampling space, which can lead to degraded performance in algorithms that operate on point clouds, such as registration algorithms, as well the acquisition of much redundant data. By performing their spherical subsampling, they ensure that the point density on the acquisition sphere is essentially uniform, and that subsequent operations performed on the subsampled point cloud are not biased based on point density. The authors' algorithm works well in the context of providing a uniformly acquired point cloud for use in algorithms that are affected by sample density bias, but it does not take into account local features of the scene, and therefore many additional points than what would be sampled via a more intelligent algorithm are acquired.

Torsello *et al.* [11] develop a method to select what the authors' term as relevant regions for the purpose of improving the performance of the registration process using the iterative closest point (ICP) algorithm [12]. But ICP tends to over fit to noise when there are large constant surfaces, such as planes or gradual curves. In order to avoid this issue, the authors resample the data such that the focus is on relevant areas, or features, such as edges and other contours. Their metric uses the concept of local distinctiveness based on the local average radius of curvature in order to determine a point representative of that local neighbourhood. Where there is deviation between these points, the region is represented by more points, and where there is no significant deviation, the local regions are merged and one of the points may be dropped if the maximum region size has not yet been attained. The algorithm successfully provides higher detailed sampling about edges and contours, and provides better results when used as part of an ICP registration algorithm. This work deals with determining where the relevant features are after acquisition, whereas the goal of this thesis is the capability of determining relevant regions to acquire with higher detail at acquisition time.

Zhu and Wu [13] have developed a method for data mining to intelligently rank fields within a database for reacquisition. Their technique identifies the importance of a field within a dataset such that database entries with that field missing can be identified, and ranked for reacquisition based upon their relative importance. They also re-adjust these rankings based

upon how certain any estimations can be made about the missing values. Their method appears to have a high degree of effectiveness across several datasets, in terms of maximizing information gain, and is interesting in the context in this thesis as the reacquisition of database field is based upon the certainty of estimates made on the missing data.

Williams *et al.* [14] have developed a methodology to intelligently acquire features using a multi-sensor platform. Since their method is for use in feature classifiers, they determine whether or not to acquire the missing point location based upon the risk of misclassification if that location is not acquired versus the cost of acquisition, i.e. the point is acquired if the cost/benefit is advantageous. They determine the value of misclassification based on the estimated benefits of acquiring the point weighted accordingly to the conditional probability of the classifier performing a misclassification based upon that point. They then iteratively apply their algorithm while acquiring new data points until their estimated risk ceases to decrease. This work is of interest for this thesis, as the cost of acquisition and the reliability of the resulting model are both considered in the acquisition strategy. Their results show that they achieve a significantly better performance when compared to random sampling approaches, but the authors do mention that their method is computationally intensive as the size of the feature classifier database increases.

Sheehan *et al.* [15] describe an automatic way to calibrate a 3D sensor comprised of three independent planar laser range finders mounted on a disk, dependent solely on acquired data. The temporal parameters (clock/skew) as well as the 3D geometry are calibrated in the technique. The interesting idea that the authors use is the concept of point cloud quality as being the goal of the calibration, i.e. the calibration procedure is considered complete when the quality cannot further improve to any significant degree. The quality parameter used is crispness, which is derived from a mixture of Gaussian model that implies that as the distribution becomes narrower the calibration becomes better. Since the authors have developed their composite sensor to be comprised of three planar laser range finders, they are able to match overlapping areas to determine feature matching, and therefore have initial estimates of the spatial calibration parameters.

English *et al.* [16] define smart sensing as a method to minimize data collection, while recognizing the object and estimating its pose. The sensor used by the authors is a laser range sensor with two degrees of freedom, due to two mirrors, which allow the variation of the azimuth as well as the elevation of the laser beam. Since the control of the mirrors is due to rotation, the optimal patterns in terms of coverage and energy use of the sensor tends toward those created by sinusoidal patterns, and they provide the three patterns that they use in their implementation: lissajous, rosette, and a spiral (Figure 2.1). The lissajous pattern is for scanning the periphery, the rosette for focusing at the center, and the spiral for uniform acquisition of features. The 'smart' sensing in their algorithm, switches between these scanning patterns based on the objects recognized in the previous scan, as well as the pose of the object with respect to the sensor. This is accomplished by first recognizing the object by an automatic target recognition (ATR) algorithm, which can identify objects using features acquired through a sparse data acquisition. A first estimate of the pose of the object is made, and refined through the use of an ICP algorithm [12]. Depending on the estimated pose, the algorithm selects an appropriate pattern such that the number of useful points acquired per complete pattern is maximized on the next scan. While this method has the benefit of providing high reliability and repeatability in its acquisition process for known and controlled environments, it appears to depend ultimately on object recognition, and the heuristic optimality of only a few patterns.

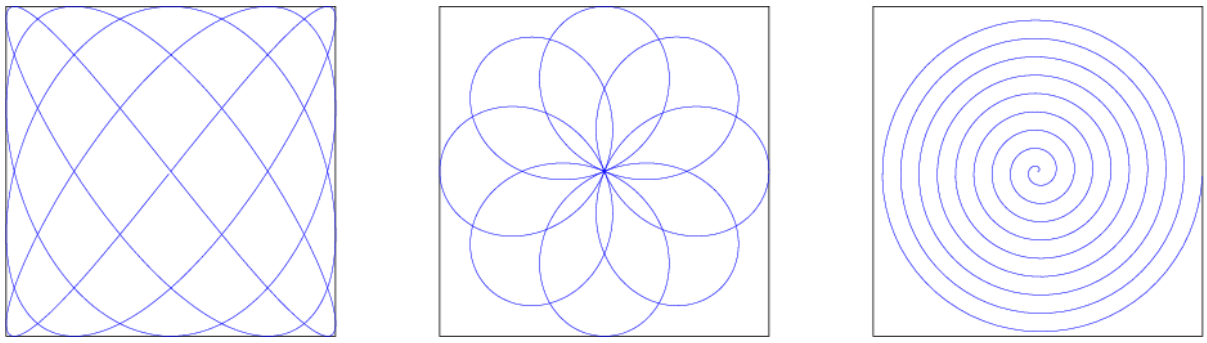


Figure 2.1 - A 4x5 lissajous pattern (left), an 8 petal rosette pattern (center), and a 9-turn spiral pattern (right)

Tsotsos *et al.* [17] present a technique which attempts to model visual attention by using selective tuning. Visual attention is the method by which animals and, with future development, machines focus on specific features and regions of a scene which are interesting, be it a goal or an obstacle. The reason why a computational solution is desirable

is that it helps minimize the quantity of data to process by providing a small number of sub images/features to work on, as well as providing greater insights into how the human visual system works. The problem of reducing a large set into small subsets of data is more similar to compression techniques, as opposed to the goal of this work, which is to build up from a coarse set to a more refined set with multiple samplings, but it is similar in terms of trying to select regions of interest intelligently. The authors discuss several ways to model visual attention, both biologically and algorithmically, as well as defining what constitutes visual attention. The algorithm that the authors propose first determines features, and then evaluates them in a hierarchal pyramid. When moving from the top of the pyramid to the bottom, sections that correspond to a feature are refined and are sections of the 'pass' zone, and the parts of the area that are not part of the refined feature, but belong to the same area that the feature belongs to at the higher level, are propagated down the pyramid in what is termed the 'inhibited' zone. This refined information is also propagated back up the pyramid to provide more detailed information in determining the importance or weight of the feature. The weight of the feature is formed from a combination of values, from the area that the feature occupies, to the intensity, colour, texture, and overall relevance to the task at hand. The drawback of this pyramid scheme is that it effectively places more weight on objects that are central in the image due to them having a larger area of influence in the pyramid when compared to objects at the edge. This corresponds well to the human visual system in that objects in the middle of the field of view tend to be the focus of the visual system. The human visual system operates over time, tends to follow moving objects, and can differentiate between a moving object and the background, which moves only when the observer changes his or her position. To account for this, the authors create an optical flow representation of the image and incorporate this into the pyramid structure, as another parameter to aid in setting weights. Additionally the human visual system tends to move between several different points over time, in what is called a saccade. The authors have built this into their algorithm by determining if there is a significant object other than the primary one currently being focused on, and then if it is significant enough to refocus the visual system onto that new object. This is done with series compensation parameters which are propagated with the features through the pyramid. The authors can achieve a similar to human oscillation of the visual system when there are multiple objects worth observing

(according to their weighted relevance). They have tested their algorithm for controlling the visual attentiveness of a robotic vision system.

In subsequent research, Bruce and Tsotsos [18] present a visual attention algorithm developed with an information theory approach. This algorithm creates a map of visual saliency, which is a map consisting of regions where the visual system is interested in looking. The map of visual saliency is constructed by first performing feature extraction. An estimate of the density of the features is made, from which a joint likelihood is produced, over which Shannon's measure of self-information is applied. In order to determine the accuracy of the proposed saliency map, the authors compare against an experiment that was performed based on the eye movements of several volunteers while they observed several different scenes. The proposed visual saliency map is found to be a fair representation of to what the human visual system pays attention. Additionally, areas with higher saliency tend to have more dominant features, which in an intelligent acquisition system, is a desirable quality. Visual attention contains interesting approaches for determining regions of interest within a scene that could be used to drive a robotic system or provide an enhanced reality perspective to human users, but it relies on processing a fully dense acquisition of a scene that is contrary to the goal of this thesis, which is to select locations during acquisition time, without much initial knowledge.

The methods presented by Cretu *et al.* [6,7], by Shih *et al.* [8], and by Torsello *et al.* [11] are the works closest to what is being proposed within this thesis. The method proposed by Cretu *et al.* [6,7] relies on the arbitrary stopping point in the training of a neural network, which can be a slow process, but works well at identifying regions to acquire at a higher resolution. It also does not identify which particular point or region would be the best choice to acquire. The first two methods proposed by Shih *et al.* [8] also identify regions that should be acquired at a higher resolution, but do so in a relatively simple way that does not determine where the best point to acquire is located. Torsello *et al.* [11] select which regions should be represented in higher detail, but only after all the data has been acquired and analysed. Visual attention as studied by Tsotsos *et al.* [17,18] is also an innovative way of focusing on regions of interest, but the methods rely on data that is already acquired at a fairly dense level, with the goal of focussed observations over time in a dynamic

environment, while the methods presented in this thesis aim at working on an initial acquisition that is very sparse. The other single point of view acquisition methods evaluate specific patterns for specific application domains, and as a result are sub-optimal for an arbitrary scene as considered in this thesis.

2.1.2 Multiple Points of View

The most popular of the decision based algorithms used in sensing is the Next Best View (NBV) class of algorithms. NBV algorithms attempt to scan an entire object by determining where the next best viewpoint to scan the object from would be, based on the past history of scanning that object. The determination of the next best viewpoint differs according to the particular algorithm. NBV algorithms are an application of active vision and path planning. The former is the coordinated movement of a sensor for acquisition at various viewpoints. The latter is the task of navigating an environment from a start position to a final position optimized around various criteria, where the primary goal is the complete acquisition of the object or scene. This section describes several methods for performing the determination of the next best view, and highlights interesting concepts in regards to this thesis.

Connolly [19] is one of the first authors to tackle the problem of where is the next best view to acquire data from. Connolly describes two methods which rely on octrees where volumes are appropriately marked as Empty, Occupied, and Unseen depending on whether the range sensor can perceive that the cell volume contains an object, or is not visible from any of the point of views currently integrated in the 3D model. The first method (Planetarium Algorithm) sets up a sphere around the scene which is evenly sampled along latitude and longitude. For each viewpoint on the sphere, a projection of the scene is determined, with the next best view being selected based upon the viewpoint which has the most unseen area. The second algorithm (Normal Algorithm) first determines the area of the unseen and empty faces in each of the six directions of the octree. The second step is to determine which areas are larger between the -x and +x facing octree areas, the -y and +y facing octree areas, and the -z and +z facing octree areas. Finally the maximal areas and their corresponding unit direction vectors are then combined to form a composite direction vector, and that composite direction vector is selected as the next best view. Both these algorithms have the advantage

that they are simple and fast, but, as the author acknowledges, do not deal well with occlusions.

Maver and Bajcsy [20] describe a method of determining the NBV by using occlusions as the primary indicator. In order to do this, calculations of where range 'shadows' occur within the existing dataset are made. The general principle behind these calculations is similar to 3-D edge detection, where localized values which are greatly dissimilar and yet in close proximity to each other indicate a possible occlusion. With the beforehand statement in mind, the authors use a polygonal representation of the acquired data, and then analyse the edges to determine if each edge is an occluding edge. Using this knowledge, a calculation of the viewing angle range for each pixel in the various polygons lying in shadow are calculated. To determine the next best view, the scanning position which optimizes the reduction of occlusions is used (i.e. the view from which most of the pixels on the polygon have been labelled as being in occlusion and share a common viewing angle). The use of calculated indicators such as possible occlusive edges is interesting and is an improvement over [19], but still has drawbacks in the sense that many precalculations must be made to set up the 3-D polygonal geometry.

Sequeira *et al.* [21] also utilize a method of view selection based on the determination of occlusions, but perform it in an active mobile platform environment. The algorithm presented uses the presence of 'double jump' edges, or in the terms of [20], polygons which are in occlusion due to a shadowing edge. The method then selects the optimal viewing angle by maximizing the number of occluding polygonal areas that will be fully revealed while minimizing the movement of the mobile platform, and while keeping in mind the safety of the mobile platform (maintaining safe distance from collisions and other danger areas). Overall, this algorithm is very similar to that of [20], but enhanced for the application of a mobile platform.

Pito [22] introduces an overlap criterion into the NBV problem. By ensuring that a certain amount of overlap is present between viewpoints, a registration algorithm can be used on the resulting acquisitions to stitch together a surface mesh of the object being acquired. Furthermore, Pito applies the concept of the visibility of a triangle when forming the triangular surface mesh of the acquired object. A triangle is given a certain visibility

quantity based on its size and orientation, relative to candidate acquisition positions, which are discretely determined as per Connolly's planetarium algorithm [19]. The selection of the NBV then proceeds similar to Connolly's method, while taking into account a minimum amount of overlap.

Zha *et al.* method [23] involves first acquiring a range image. The next best view is then determined by the amount of desired overlap, the desirability of exposing occlusions, and the constraints on maintaining the smoothness of the model all based upon the reachable 'view' space of their range sensor. This type of approach, while ensuring that the registration of data is possible (in case sensor position cannot be relied upon) does involve many redundant sampling points. The performance optimization of this technique, as presented by the authors in [24], is made by discretizing the possible view space of the view sphere, which reduces the number of computations required in calculating the NBV. The addition of a discretized view sphere leads to the possibility of using lookup tables, and as such eliminates many recalculations that would otherwise be necessary after new data is added to the model, as only the new data needs to be added to the calculations. The speed using lookup tables will be faster than the speed of performing run-time calculations of the same data, but there will be the possible reduction in quality by not utilizing a more continuous view sphere that may be acceptable depending on the application. The usage of both overlap and occlusions within the space where the sensor can reach as parameters for guiding the NBV process is interesting as the resulting system could potentially be operated using solely the acquired data, and not on any other localization sensing methodologies, at the cost of acquiring much redundant data.

Reed *et al.* [25] describe a method for range image acquisition planning for the purpose of creating CAD models. Their algorithm operates by acquiring an image, and then stitching together surface meshes from the discrete data points. These surfaces are then extruded to create solids, and then each surface is labelled as visible or occluded based upon the length measurements of the edges and whether the sensor actually saw the surface directly. The next best view is then determined using a planetary method similar to [19], using ray tracing from each planetary view to determine which view will reveal more of the labelled occlusions of the model. A new model is created using geometric primitives of the most

recent scan to alter the existing model, and the process is repeated. This algorithm appears to do a good job at recreating a viable 3-D model of the object being scanned, but requires several dense scans and complex 3-D intersection calculations.

Wong *et al.* [26] introduce a baseline method in order to evaluate the effectiveness of NBV algorithms, and then compare it to their own method, as well as a hybridized method. The baseline method, which the authors term 'The optimal method', essentially brute force calculates the amount of unknown voxels for all viewpoints on a discretized view sphere, which is similar to what was performed in [19,24,25]. The authors then present what they term 'the surface normal method', in which they determine the next best view by adding together the surface normals of each side of each unknown voxel. This second method tends to not do well with self-occluding objects, so the authors merge both methods together to create 'the adaptive method' which first does the surface normal method, and switches to the optimal method when self-occlusions occur. The combination of the two proposed methods into a single hybridized method in order to enhance the overall performance of the NBV process, while setting the ground work for evaluation of NBV algorithms makes this work significant for this thesis.

Klein and Sequeira's work [27] is different from the previous works, as they use a quality measure and a movement penalty in order to help determine the NBV while still allowing it to be practical for large scale environments with relatively slow moving sensor platforms. The first step that they performed is the scan, to which they then apply a triangular mesh, as well as zippering together meshes of previous stages, favouring meshes of higher density acquisitions. Any surfaces that are not resolved, due to occlusions or range, are labelled as void surfaces. An objective function then calculates the best position to reveal void surfaces based upon the field of view and the viewing direction of the sensor. Additionally, the metrics of the quality of the acquired data (density of acquisition), and the cost of each scan (to avoid close in acquisitions of high density, but low overall revealing of void surfaces) are taken into account. To do this in an efficient manner, the authors have taken full advantage of current graphics technologies, mainly the use of z-buffers in OpenGL. After acquisition, the scene is rendered in OpenGL, and then perspective 'snap shots' from all six sides of the view cube are acquired, with one of the RGB components representing sampling density.

The z-buffer and sampling density maps are then used in the objective function to simplify calculations.

MacKinnon's work [28,29,30] is a variation of the next best view problem. He redefines the problem as to where to scan next with a laser range finder based upon several quality metrics. The quality metrics that he uses are defined as resolvability, planarity, orientation, reflectivity, and spot size. Using these metrics MacKinnon is able to ensure that areas acquired at oblique angles, at low resolution, or with high variability are rescanned from a more optimal pose. By iteratively merging the new data acquired from different more optimal angles, a more efficient and complete acquisition of an object can be performed, especially compared to a brute force approach. MacKinnon's method is strongly tied to a specific type range sensor that will provide the needed values for the quality metrics during acquisition.

The art gallery problem [31] is also a variation of the NBV problem, where the floor plan of an art gallery is presented as a 2D polygon. The task which is to be performed is to place 'guards' on the floor plan such that all the walls can be seen by the guards. This is equivalent to the application of NBV where a model of the object is known *a priori*, but a detailed inspection must be performed for defect analysis. González-Banos and Latombe [32] propose a strategy to solve the art gallery problem by selecting random guard positions, and then prune unnecessary candidate positions based on set coverage. The authors present a greedy algorithm as well as an elastic algorithm for the pruning process. Taking a random subset is a common practice in computer vision (e.g. RANSAC algorithms for identification and registration). The drawback to this sort of random position selection is that it is possible to leave certain elements without any coverage if the randomization is performed naïvely, or if the algorithm does not adjust itself by adding more points if it detects incomplete coverage. The advantage of randomized sampling techniques is that it reduces the number of positions which need to be evaluated, thereby reducing the computational burden of the final solution compared to the naïve brute force solution.

The NBV algorithms presented in this section are applicable to building up models by choosing where to scan next in a global sense, but are not directly suitable to the problem of minimizing the number of data points acquired within a single point of view. That being

stated, they do highlight several issues, namely those of non-visible regions (occlusions, void spaces) by the minimization of unknown values based upon occlusions and edges. Furthermore, as NBV tends to be computationally intensive, many methods impose restrictions to reduce candidate viewpoints or apply approximation techniques in order to reduce the number of required computations. The key problem that NBV tries to solve is the problem of filling in missing information, but NBV does it on a very large scale (i.e. determining the complete 3D surface model of an object) which differs significantly from the goal of this thesis.

2.2 Estimation Techniques

Estimation techniques use known values, which are values at locations that have been measured, to provide an estimate of unknown values, which are values for locations that have not been measured. Estimating unknown values within a region bounded by the set of known values is called interpolation, while estimating unknown values outside of the boundary of the set of known values is called extrapolation. There are several different ways to accomplish these tasks. There are piecewise methods, which break down the problem into several small problems, where the solution is a simple shape or curve which fits to the localized existing data points, and there are global based solutions which attempt to fit a curve or a shape to the dataset as a whole. Moreover there are non-parametric methods that use the data directly without the inference of any particular trends to infer missing data. The drawback to any estimation methods is the level of uncertainty in the estimated points - the unknown datum cannot be known exactly without sampling the point directly, and even sampling the point directly has error involved in the process. Furthermore, estimation techniques can suffer from various drawbacks, such as over fitting and under fitting [33].

Over fitting includes the fitting of the noise with the underlying measured data, and as a result any estimation will include the parameterization effects of the noise present in the underlying data. Over fitting is usually the result of modeling the trends with too many parameters. Under fitting overly smoothes the data to the point where the estimation error becomes large, and is a result of the lack of sufficient quantity of parameters in the estimation model. The goal of any estimation technique is to optimally provide an estimate

of the unknown values and to minimize the possible occurrence of over fitting and under fitting.

There are several techniques which one can use to estimate new values based on existing known values. Curve fitting attempts to fit a parameterized function, or linear combination of basis functions to the dataset. For example, the Fourier transform can be used to fit a series of orthonormal complex exponential basis functions to a regularly sampled dataset to approximately reconstruct the source signal. Other curve fitting methods use stochastic parameter optimization in order to fit known data points to various trend models, such as polynomial, logarithmic, and exponential functions. The most common method of fitting curves is least squares optimization, which attempts to minimize the error of the estimated value with respect to the parameters of the functions.

Another commonly used function in curve fitting is linear regression. Linear regression fits a line to within a known set of data, while minimizing the mean squared error of the known values. Higher order polynomials are also used, but as the order of the polynomial approaches the number of known data points, the chance that the estimation will be an over fit increases, conversely a linear estimation has a tendency to under fit the data, but could give good insight as to the overall trends in the data. Other, more specialized methods to aid in estimation of values exist.

In the following subsections, approximations will be discussed, as well as the theory behind approximations. Following that, a brief discussion on function estimators is made, containing an evaluation of the estimation errors, and the applicability to the context within this thesis. Splines, which are common in computer graphics, are presented, followed by neural networks and their applications, which are relevant to this work, are discussed. Estimating physical measurements via fitting a mathematical function is not the sole method to estimate unknown values, as the fifth subsection discusses applications in data mining. Afterwards Kriging, an estimation technique which minimizes the estimation variance is discussed and developed, as well as the supporting statistical technique of semivariograms. Finally the more generalized Gaussian process is discussed.

2.2.1 Approximations

Approximations allow us to roughly characterize systems and physical processes in order to make simplifications such that the problems become more tractable to analysis [33]. Approximations also aid us in modelling systems and calculating values for which complete knowledge is unavailable, but which is known to be close enough that only small errors are observable. As such, interpolations and extrapolations involving data points can yield only approximations of the underlying behaviour of the systems involved in producing the known datasets.

There should be care taken in making approximations, as approximations typically have only a range of support, beyond which the approximation error starts becoming significant depending on the application [34]. The approximation theorem, shown in eq. (2.1), states that for a function, $f(x)$ that is continuous from a to b, there exists a polynomial, $p(x)$, that approximates it with a very small acceptable error, ε. By extension, interpolation techniques just approximate a value that exists between data points.

$$|f(x) - p(x)| < \varepsilon, \qquad a \leq x \leq b \tag{2.1}$$

2.2.2 Functional Estimators

Herzberg and Cox [35] discuss designs of polynomial interpolators and extrapolators to minimize the effect of random error. The authors make the assumption that the process used to generate the data points have the same variance, and that any two data points are uncorrelated. The primary approach that the authors discuss is the Lagrangian interpolation polynomial, with the approach objective being to minimize the variance of the estimated values obtained from the polynomial. The authors show that as the degree of the polynomial increases, so does the variance of the extrapolator, and that an optimal variance can always be determined for an interpolator. Additionally, the authors provide formulae for determining the variance at any other point, provided that one knows the variance at the centroid, as well as the inherent variance of the process which produced the data. By inspecting the variance equation, it is observed that by minimizing the inverse of the correlation matrix of the independent variables to be as small as possible leads to a minimization of the variance at the estimated point. Unfortunately, the techniques discussed rely on knowing the stochastic process behind the data producing techniques to within

acceptable tolerances, which is not usually available in uncontrolled arbitrary scenes, which is the scenario being explored for this thesis.

2.2.3 Splines

Splines are piecewise curves which attempt to minimize hard edges where different piecewise components meet, and they can be used on regular, as well as irregular grids [34,36,37,38]. Splines are used in many applications from geophysics to computer graphics, and are extremely popular interpolation tools due to their relative simplicity, and low computational resource needs. Equation (2.2) details the general system of equations for a typical spline, $g(x)$, namely that it is a piecewise curve, and that at the points that the curves meet, the piecewise segments must have an equal value, and that its first and second derivatives must be equal.

$$\begin{gathered} g(x) = \begin{cases} g_1(x), & x_0 \le x \le x_1 \\ g_i(x), & x_{i-1} \le x \le x_i \\ g_n(x), & x_{n-1} \le x \le x_n \end{cases} \\ g_i(x_i) = g_{i+1}(x_i) \\ g_i'(x_i) = g_{i+1}'(x_i) \\ g_i''(x_i) = g_{i+1}''(x_i) \end{gathered} \tag{2.2}$$

There are several different types of common splines: Bezier Splines, B-Splines, Cubic-Splines, and more. Moreover, there exists splines that are defined by behaviours other than that defined by eq. (2.2). Each type of spline has a different characteristic function and behaviour, as well as computational requirements.

2.2.4 Neural Networks

Neural networks are an attempt at modelling phenomena based upon the mimicry of the neurons located in the human brain and nervous system [39,40,41]. Neural networks have been successfully used to solve linear and non-linear problems, including system identification and approximation. But the real interest in neural networks is modelling complex problems whose input and output relationships are not readily defined using traditional methods.

Neural networks are composed of neurons which accept a series of inputs, which are added together with a bias coefficient as a weighted sum, and then passed into an activation function to produce an output (see Figure 2.2). The neurons can be arranged in cascade for

multiple levels, with each neuron in the next level having inputs corresponding to all the outputs of the neurons in the previous level, with the possibility of feedback and feed-forward connections. The multiple level, or layers, effectively let the network capture more of the statistical properties of the input signals. In order for a neuron, and in extension, a neural network, to be useful, the weights of each neuron need to be trained. This can be accomplished through several means depending on the task the neural network is to solve, the network structure of the neurons, and the activation function employed. Training a neural network can be performed offline, or online in certain scenarios.

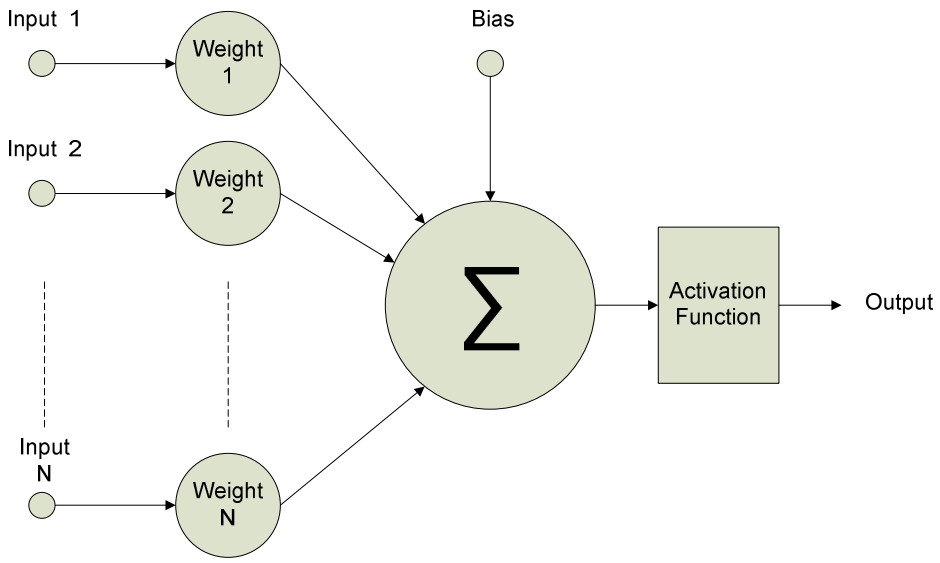


Figure 2.2 - Diagram of a neuron

Haykin [39] provides several examples of neural networks being used as classifiers for object recognition, speech recognition, compression of signals, and reconstruction of signals in addition to discussing issues related to scaling of the input signal (such as larger/smaller images and faster/slower speech). For interpolation tasks, Haykin states that the radial basis function (RBF) based network is the appropriate neural network for identifying and interpolating functions. The RBF network contains one input layer, one hidden layer, and one output layer. The input corresponds to the input parameters, and the output corresponds to the result of the function. Typically an RBF network is trained offline, but with changes to the training algorithm, online training could be used.

Principle component analysis (PCA) can also be performed by neural networks [39,40], which can allow for approximations as well as compression of signals. PCA allows the determination of dominant and non-dominant trends of the input signal such that the dominant influences can be extracted and analysed. In compression applications, dominant components are quantized with higher number of bits, while non-dominant components can be dropped, or characterized more coarsely with a lesser number of bits. Similarly in approximating missing samples, or filtering a noisy signal, the most dominant components can provide the trending information for a possible reconstruction of the underlying signal.

Neural networks can also be used to make decisions, such as classifying objects, and performing action/reward based decision analysis as what is used in decision processes [39,41]. The fact that neural networks can approximate complex functions makes them a useful tool, but unfortunately the problems of effectively training them, and updating the weights introduces issues that could produce a more complex and less efficient solution for more common problems that are already solvable using a direct linear system. Additionally, due to the large number of parameters inherent in neural networks, there is the risk of over fitting [42]. Where the underlying model is unknown, or complex, neural network have shown to provide good performance.

2.2.5 Data Mining

Data mining is a process where one goes through one or more databases looking for statistical properties and linkages. As with any database, there will be missing fields, duplicate fields (e.g. entries of P. Curtis, or Phillip Curtis, followed by essentially the same information), or incorrect fields, which are synonymous with noisy or corrupt data in the fields of signal processing. Methods exist to try to interpolate between multiple databases and merge them together, similar to registration and data fusion in imaging.

Zhu and Wu [13] have developed a method to fill in missing fields within a database through what they call selective data acquisition. They base their method on three goals: more attention should be paid to fields that have high correlation to what is being searched, to ignore empty fields that do not apply to the model for which the data mining is being performed as it will yield little in return, and that some missing fields may be predictable through other fields. The first step the authors perform is to try to find important fields

within the dataset through correlation techniques. A data field with a higher correlation has a higher importance and gets assigned what the authors term an 'information gain ratio'. Entries within the database which contain empty fields corresponding to high information gain ratios get selected as entries that should be reacquired first. The second step is to determine whether for the entries selected to be reacquired, if an estimate can be determined with a high degree of certainty, then the priority of reacquisition for that particular entry would be lowered appropriately. This technique balances the conflicting desires of filling in high priority information areas, as well as making sure that reacquisition of data does not occur where a reasonable approximation can be made.

2.2.6 Kriging

Kriging is a stochastic estimation technique which minimizes the variance of the estimated value, and is an exact interpolator, meaning that an estimate on a known value, will return that value [43]. Kriging is used extensively in the geosciences for estimating gradient fields for geography [43,44,45], expected crop yields [46], expected ore deposit concentrations, and more. It does not require the known data points to be regularly sampled, but relies on the principle that as one gets further from known points, then the less similar to those known points the data value will become, i.e. it relies on the correlation based on locality of the data.

Dunlap and Spinazola [47] use Kriging to interpolate water-table altitudes in West-Central Kansas, and choose Kriging since it does not require a regularly spaced lattice like many of the other choices (such as polynomial fitting as in Lagrange interpolation which tends to yield inaccurate results when higher order polynomials become involved). They also found that in weighted-interpolation techniques, the major parameter was the distance of the point to be estimated, with no consideration for direction, as well as areas with a higher density of points get effectively weighted higher than nearby sparser groups of points. Least squares (LS) approaches calculate a surface for the data, which may not pass through the set of measured points, meaning that LS approaches are inexact interpolators, as well as behaving on a global scale instead of considering local variations. Dunlap and Spinazola choose Kriging for four reasons: its ability to estimate the reliability of data at unmeasured sites, it is an exact interpolator at measured sites, it incorporated the local stochastic properties of the sites being estimated, and it is reproducible at estimating values at unmeasured locations.

Additionally in the computer code that they used to produce their results, they only used local neighbourhood points, since in Kriging, points that are further away will have weights that are very small compared to more local points. They use the Kriging error to analyse where more measurements need to be taken to improve the results of the overall measuring of the altitude of the water table for future studies.

Kriging has been used in the field of 3D image processing outside of geo-spatial applications. Grinstead *et al.* [48] use Kriging for feature enhancement and denoising 3D range images. The authors acknowledge that global optimization using Kriging overly smoothes the data, just like other stochastic techniques such as Wiener filters, and so they limit themselves to local regions. The local regions are determined by segmenting a 3D triangular mesh based upon connected regions of triangles displaying similar curvature profiles. Anything that is contained in a region, but that is not within a region displaying similar curvatures (i.e. highly textured areas/regions of high variability), are selected as local regions to be enhanced through their Kriging based technique. The results that have been obtained tend to match their ground truth models better than comparative methods such as linear interpolation, cubic interpolation, and inverse distance weighting interpolation, but at a much higher computational cost.

Ho and Saripalli [9] use Kriging to interpolate between sampled points within a scalar field for the mapping of data acquired in AUV sensing applications in order to represent the entire environment. They use the information in regards to Kriging error to infer accuracy of their acquired data to help compare several acquisition patterns in terms of efficiency of power consumption for coverage and accuracy.

Haas [49] develops a technique to minimize the computation time, while increasing the effect of locally sampled points in the Kriging calculations by introducing an automated variogram modelling technique using a moving window. Haas mentions several advantages to a moving window: that the number of points used in the calculations is lessened, any effect of non-stationarity trends in the data will be minimized by taking only local regions, and local effects are better characterized. A circular window is used, which is centered on the point to be estimated. The circle is enlarged until certain criteria are met, with the main one being that at least 35 sampled sites are contained within the window. A semivariogram

model (which is chosen *a priori* based upon known behaviours of the properties being measured) is then fit using a least squares method to the empirical semivariogram data. The result of the work is an increased accuracy and confidence compared to the ground truth in predefined sub regions, at the expense of increased computational time (due to having to calculate many more semivariogram models).

Several variations of Kriging exist depending on how much is known or assumed about the general trends of the data. Simple Kriging assumes a constant mean, Ordinary Kriging assumes a varying mean that is more or less constant in a local neighbourhood, and Universal Kriging (also known as Kriging with a trend) assumes that the mean varies according to some kind of trend [43]. Under assumptions of simple Kriging and Ordinary Kriging, a symmetric covariance is typically used due to their relative simplicity depending upon the phenomenon being modelled, with a radially symmetric covariance being used where no in depth knowledge of the system is known *a priori*. For the case of Universal Kriging, a directionally dependant covariance function may be desired to better model the various trends.

Ordinary Kriging is the type that will be used in this thesis, mainly because in range imaging it is usually safe to assume that the underlying depth observed within a localized region is going to be fairly constant within that region, even though it will vary globally. Within this thesis for the sake of convenience, when the technique Kriging is mentioned, the reader can assume that Ordinary Kriging is the specific technique unless otherwise mentioned. The Kriging estimator function, using the same notation as what will be used later on in this thesis, is written as:

$$\hat{z}(\hat{p}_j) = m(\hat{p}_j) + \sum_{i=1}^{n} \lambda_i(\hat{p}_j)\left[z(p_i) - m(\hat{p}_j)\right] \tag{2.3}$$

where $\hat{z}(\hat{p}_j)$ is the estimated value, $\hat{p}_j$ is the location where the estimation is being performed at, $m(\hat{p}_j)$ is the local mean, n is the number of known points, $\lambda_i(\hat{p}_j)$ are the 1 to n weight factors, and $z(p_i)$ is the ith known value at location p_i. By rearranging the equation, and enforcing the criteria that the Kriging weights must sum up to one, the resulting system of equations becomes:

$$\hat{z}(\hat{p}_j) = \sum_{i=1}^{n} \lambda_i(\hat{p}_j) z(p_i) \quad , \sum_{i=1}^{n} \lambda_i(\hat{p}_j) = 1 \tag{2.4}$$

If the weights are then calculated in order to minimize the estimated variance, then the system of equations for the Kriging weights become:

$$\begin{gathered} \sum_{i=1}^{n} \lambda_i(\hat{p}_j) C(p_k - p_i) + \mu(\hat{p}_j) = C(p_k - \hat{p}_j), k = 1, \dots, n \\ \sum_{i=1}^{n} \lambda_i(\hat{p}_j) = 1 \end{gathered} \tag{2.5}$$

where $C(h)$ is the covariance function (with $C(0)$ being the covariance function at zero distance), and $\mu(\hat{p}_j)$ is a Lagrange parameter. Once the Kriging weights are determined, the Kriging variance is represented by the following equation:

$$\hat{\sigma}^2(\hat{p}_j) = C(0) - \sum_{i=1}^{n} \lambda_i(\hat{p}_j) C(p_i - \hat{p}_j) - \mu(\hat{p}_j) \tag{2.6}$$

The semivariance, $\gamma(h)$, is a measure of how a set of values varies over distance. As this is easier to calculate directly from raw data than the covariance, it is used instead. If the covariance, $C(h)$, exists at a certain distance h, then the semivariance is related to the covariance by the following expression:

$$\gamma(h) = C(0) - C(h) \tag{2.7}$$

By substituting in the semivariance for the covariance using eq. (2.7) into eq. (2.5) and (2.6), then the Kriging system of equations for weights becomes:

$$\begin{gathered} \sum_{i=1}^{n} \lambda_i(\hat{p}_j) \gamma(p_k - p_i) + \mu(\hat{p}_j) = \gamma(p_k - \hat{p}_j), k = 1, \dots, n \\ \sum_{i=1}^{n} \lambda_i(\hat{p}_j) = 1 \end{gathered} \tag{2.8}$$

And the Kriging variance becomes:

$$\hat{\sigma}^2(\hat{p}_j) = \sum_{i=1}^{n} \lambda_i(\hat{p}_j) \gamma(p_i - \hat{p}_j) + \mu(\hat{p}_j) \tag{2.9}$$

Converting the values to matrix notation, where

$$\lambda(\hat{p}_j) = \begin{bmatrix} \lambda_1(\hat{p}_j) \\ \lambda_2(\hat{p}_j) \\ \vdots \\ \lambda_n(\hat{p}_j) \\ \mu(\hat{p}_j) \end{bmatrix} \tag{2.10}$$

$$K = \begin{bmatrix} \gamma(p_1 - p_1) & \gamma(p_1 - p_2) & \dots & \gamma(p_1 - p_n) & 1 \\ \gamma(p_2 - p_1) & \gamma(p_2 - p_2) & \dots & \gamma(p_2 - p_n) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(p_n - p_1) & \gamma(p_n - p_2) & \dots & \gamma(p_n - p_n) & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \tag{2.11}$$

$$k(\hat{p}_j) = \begin{bmatrix} \gamma(p_1 - \hat{p}_j) \\ \gamma(p_2 - \hat{p}_j) \\ \vdots \\ \gamma(p_n - \hat{p}_j) \\ 1 \end{bmatrix} \tag{2.12}$$

$$Z = \begin{bmatrix} z(p_1) \\ z(p_2) \\ \vdots \\ z(p_n) \\ 0 \end{bmatrix} \tag{2.13}$$

Yields the Kriging matrix formulation:

$$\begin{gathered} K\lambda(\hat{p}_j) = k(\hat{p}_j) \\ \lambda(\hat{p}_j) = K^{-1}k(\hat{p}_j) \end{gathered} \tag{2.14}$$

$$\hat{z}(\hat{p}_j) = \lambda^T(\hat{p}_j)Z \tag{2.15}$$

$$\hat{\sigma}^2(\hat{p}_j) = \lambda^T(\hat{p}_j)k(\hat{p}_j) \tag{2.16}$$

2.2.7 Semivariogram

The semivariance, $\gamma(h)$, as shown in eq. (2.17), is a measure of half the mean squared difference between measured values at a displacement of h, where $n(h)$ is the number of point pairs with displacement of h, and $z(p_i)$ is the measurement at point p_i. The semivariogram is an ordered graph that maps displacements between many measurements, to the respective semivariance of those measurements. The term semivariance is used when referring to a particular value, and semivariogram is used when talking about the overall mapping function.

$$\gamma(h) = \frac{1}{2}\frac{1}{n(h)}\sum_{i=1}^{n(h)}\big(z(p_i + h) - z(p_i)\big)^2 \tag{2.17}$$

$$\gamma_{emp}(h) = \frac{1}{2} E\{|z(p_i + h) - z(p_i)|^2\} \tag{2.18}$$

An empirical semivariogram, $\gamma_{emp}(h)$, is determined by calculating the pair-wise displacement between the measured locations contained in the dataset, and then determining the expectation of semivariances for each particular displacement. Note that since the empirical semivariogram relies on calculating the distances as well as the expected semivariance between all pairs possible between the measured data points, there will be a total of $\frac{n!}{2!(n-2)!}$ possible pairs for n points. Figure 2.3 shows an example of an arbitrary function, $z(x)$ and its corresponding empirical semivariogram.

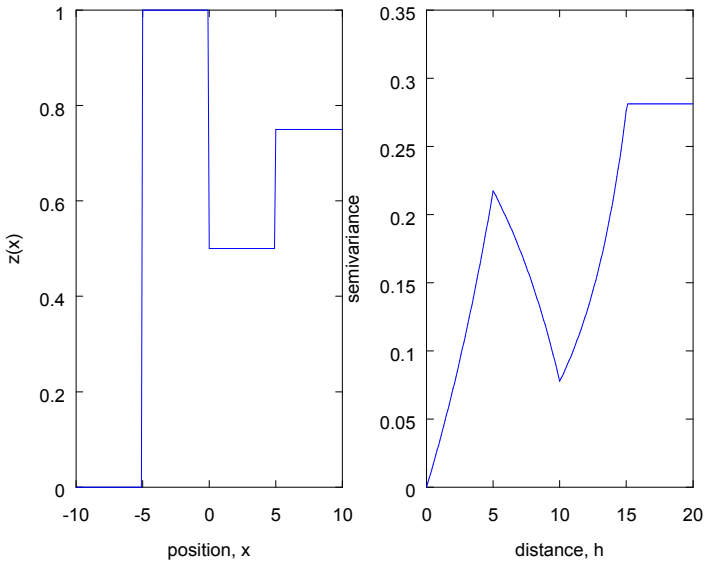


Figure 2.3 - An arbitrary function, z(x) on the left, and its empirical semivariogram, right.

As can be seen in Figure 2.3, an empirical semivariogram of an arbitrary function, when plotted, will not generally be a smooth curve and due to this reason it is preferred to fit a function to the empirical data [44,50,51]. Not all functions can represent a valid semivariogram, and a valid semivariogram is termed a permissible semivariogram. The properties which define a permissible semivariogram function [44] are:

1. The semivariogram is non-negative, $\gamma(h) > 0$
2. The semivariogram is zero at zero, $\gamma(0) = 0$
3. The semivariogram is conditionally negative definite,

$$\sum_{i=1}^{n}\sum_{j=1}^{n} w_i \gamma\left(h(p_i, p_j)\right) w_j \leq 0, \text{ iff } \sum_{j=1}^{n} w_j = 0$$

Fortunately several semivariogram models have already be determined as permissible models (see Table 2.1), and any linear combination of permissible models produces a permissible model [44].

Semivariogram Model	Mathematical Model
Nugget effect model	$\gamma(h) = \begin{cases} 0, & if\ h = 0 \\ 1, & otherwise \end{cases}$
Spherical model with range a	$\gamma(h) = \begin{cases} 1.5\frac{h}{a} - 0.5\left(\frac{h}{a}\right)^3, & if\ a \leq h \\ 1, & otherwise \end{cases}$
Exponential model with range a	$\gamma(h) = 1 - e^{\frac{-3h}{a}}$
Gaussian model with range a	$\gamma(h) = 1 - e^{\frac{-3h^2}{a^2}}$
Power model	$\gamma(h) = h^w, \quad 0 < w < 2$

Table 2.1 - Common semivariogram models

The fitting of the semivariogram model to the empirical semivariogram is accomplished using various techniques, from maximum likelihood models [51], to least squares and parametric curve fitting [49].

The semivariogram is used extensively in the Kriging process, and the Kriging process requires that the semivariogram model used is permissible, otherwise the matrices involved can be ill-conditioned or even singular.

2.2.8 Gaussian Process

Williams [42] shows that Bayesian linear regression can be represented as a Gaussian predictor process. He then expands upon this work to show that predicting using a Gaussian process is also related to certain neural network models, Kriging, and splines. Neural networks, specifically the RBF neural network, can be modelled directly as a Gaussian process when the covariance function depends solely on the Euclidean distance. Kriging is a Gaussian process as are splines, which can be modelled as Gaussian processes with a specially chosen covariance function. All of these can be used in a linear fashion to both interpolate, and to predict values. By knowing how the different types are inter-related, it is

possible to switch between implementations if it is desired, Gaussian processes like Kriging however directly provide an estimate of the variance of the measure, while the naïve approaches to RBF neural networks and splines do not.

2.2.9 Critical Analysis of Estimation Techniques

In the context of this thesis, which is to progressively determine the best location where to acquire the next range measurement, based upon the previously acquired data from a single point of view, an estimated depth for all unknown samples that are acquirable for the sensor is desired, as well as an estimate of the magnitude of error for the depth estimate. Additionally, a local approach is desired over that of using a global approach, as there may be several different regions in the range image that exhibit different stochastic properties.

Functional estimators, while providing both an estimate and an estimated variance, require direct knowledge of the underlying stochastic process to perform well. As acquiring a range image of an unknown environment involves different possible regions exhibiting unknown stochastic properties, function estimators are not a good approach for this application. Splines, due to their piecewise nature, will produce localized results and they do not require any knowledge of the underlying surface. The drawback to splines is that they do not directly produce an estimation of error in the estimation. Neural networks, and in particular RBF neural networks work well for estimating values, but have a similar drawback to splines in that they do not implicitly generate an estimate of the estimation error. Ordinary Kriging fits the goals of producing both an estimated value, as well as an estimation of the estimation error. Additionally it has been shown to work well with localized point selection, and by using a semivariogram, the local variation can be effectively captured within the estimation. Finally, as Kriging is a type of Gaussian process, as are splines and RBF neural networks, ideas from both can be used to influence the design of the final algorithm, as well as providing the possibility for any future implementations to switch to a different Gaussian process.

2.3 Decision Processes

The work in this thesis is to intelligently decide where to scan the next point, based upon the acquisition of all previous points, while balancing the possibly conflicting demands of scene coverage and model accuracy. This process of deciding what to do based on a goal and a set

of existing data is what is termed as a decision process. Decision processes take data from one or many data sources to guide a decision, usually based upon the optimization of some set of parameters and involving probabilities of outcomes as well as quantifying the rewards of beneficial outcomes. Most methods use some sort of heuristic analysis to rank possibilities and their respective rewards, and then select the most appropriate action to perform. In this section, a variety of techniques will be discussed, starting with the classical Markov decision process (MDP) and continuing with the partially observable Markov decision process (POMDP). Following that, a look towards some applications of these techniques as well as some computationally more efficient heuristic techniques and adaptations of these methods will be presented. Finally, a brief look at the applicability of neural networks to a decision process is performed.

A Markov decision process (MDP) is a technique which applies optimal decision making in order to execute a set of actions to reach a goal for a Markovian process [52,53,54,55]. An MDP has several parameters associated with it, as illustrated in Table 2.2.

Parameter	Meaning
$\boldsymbol{S}$	Set of s States
$\boldsymbol{A}$	Set of Actions based on the set of states
$\boldsymbol{P(s,a,s')}$	Probability that an action a at state s at time t, will result in state s' at time $t+1$
$\boldsymbol{R(s,a,s')}$	The reward for transitioning between state s to state s' after performing action a
$\boldsymbol{V(s)}$	The value associated with the state s.
$\boldsymbol{\alpha}$	Future discount factor

Table 2.2 - MDP parameters list

The optimal solution for the problem of maximizing the value of an action based on the current state transitioning to a new state is shown in eq. (2.19). The central concept behind this equation is to weigh the expected rewards of the state transition with the discounted future value that will result from the transition to a new state, s', by the probability of that transition. By discounting the future value, there is more emphasis placed upon the reward of the action, as opposed to the future value, which may be uncertain due to a dynamic environment. These conditions combined yield a choosing of the action that will yield the

most benefit to the current state, while still allowing the consideration of maximizing the future value, and while realizing a great benefit to the current instant in time.

$$V(s) = \max_{a \in A} \left\{ \sum_{s' \in S} P(s,a,s')[R(s,a,s') + \alpha V(s')] \right\} \tag{2.19}$$

When the states cannot be observed directly, then a Partially Observable Markov Decision Process (POMDP) can be used instead [54,55]. With a POMDP, one still has all the parameters associated with an MDP, but the additional parameter of observation is added. The observations allow the modification of the probabilistic belief state which corresponds to the probability that any state is the current state. The way that a POMDP works is that based on the assumption of a current belief state, then the action a is made, followed by a series of observations which allow the updating of the belief state. The parameters added to a MDP for a POMDP are described in Table 2.3. The belief update is shown in eq. (2.20) (where η is a normalization factor) and the new value formula is shown in eq. (2.21). A POMDP yields the optimal action for each possible belief over the all the possible states, and hence for large problems, can become computationally intractable due to the exponential growth in calculations and memory with respect to the number of states.

Parameter	Meaning
$\boldsymbol{O}$	Set of Observations
$\boldsymbol{b_t(s)}$	Belief of state s at time t
$\boldsymbol{\Omega(o,s,a)}$	The probability of observing o from state s after performing action a

Table 2.3 - Added parameters for POMDP

$$b_t(s') = \eta\, \Omega(o,s',a) \sum_{s \in S} P(s,a,s') b_{t-1}(s) \tag{2.20}$$

$$V(b) = \max_{a \in A} \left\{ \sum_{s' \in S} P\big(b_t(s), a, b_t(s')\big) \big[R\big(b_t(s), a, b_t(s')\big) + \alpha V b_t(s') \big] \right\} \tag{2.21}$$

Cassandra *et al.* [56] discuss various techniques for deciding how to navigate a mobile robot using discrete Bayesian models. The authors develop the optimal control strategy for a mobile robot in a probabilistic reality using a POMDP, and then develop several heuristic control strategies and compare the various techniques. The first step for developing a POMDP for a mobile robot is to define the mapping of the environment, the state that the

robot is in (position, or grid cell, and orientation), as well as the actions that can be performed. The authors choose to discretize the environment into 1 m^2 cells, limit the robot to the four cardinal directions, and provide 5 actions being 'move-forward', 'turn-left', 'turn-right', 'no-op', and 'declare-goal'. From the current state information, and the actions that can be applied, a probability that a future state is reached is determined. For example if the robot is facing a wall, the probability that moving forward will yield a new outcome is low, because the robot is blocked by the wall. Using a POMDP, one determines the conditional probability of an observation given the current state and a set of actions, or in other words what is the probability of a future outcome if the robot is in a particular state and performs a set of actions. The authors use a probabilistic occupancy grid to integrate sensor data into the robots awareness of its environment. Since the robot is never entirely sure where it is, its belief in its state (position and orientation) is probabilistic, if it was deterministic, a regular MDP could be used in place of a POMDP. Since the probability is going to be spread out in space, there will be a set of belief states, each with a probability of the robot believing that it exists within that state. The optimal decision on where to move is based on a reward function that takes into account the benefit that the robot has for applying an action provided that it is in the current state. By propagating this information into the future for the robot to reach its goal, a value is assigned based upon the propagation of states, and the expected rewards. Future estimated states, rewards, and actions are given lesser weight the farther into the future they are, as uncertainty increases. The final optimal path is the one with the highest value, and it is the one chosen to act towards in the next action. Continuously calculating all future steps, as well as calculating the rewards and path values after each action is computationally expensive, and in the case of large grid (which yields a large number of belief states), intractable to do in a reasonable amount of time. As a result, heuristic approaches are often preferred.

The three heuristic algorithms that Cassandra *et al.* propose in order to alleviate some of the drawbacks of the optimal POMDP are discussed in the following paragraphs. The first method is the Belief Replanning heuristic algorithm, where an assumption on the most likely world state is made, and a path planning assuming a deterministic idealization of the path through the environment is also made. During this planning phase, each expected observable outcome along the path is determined, and then the robot progresses down the path. When

the robot follows the path, it updates its belief state and checking at each step that the world state matches its predicted state. If it does not, the planning process is re-executed, and the process begins anew. The difference between this and the original POMDP is that the Bayesian belief state is not computed, but only observational uncertainty is used to compute certainty factors on the world states.

The second method presented applies an MDP instead of a POMDP, which is much faster to compute. The way that the technique works is that the most likely world state (MLS) is chosen as the current state, and then applies an MDP to plan the optimal course of actions. This is more thorough than the previously discussed method of replanning, as it takes into account more probabilistic outcomes.

The third and final method discussed by the authors, uses the degree of uncertainty as the guiding force in deciding what path to follow. The POMDP that the optimal path planning uses balances the path plan based upon reducing uncertainty of states, as well as driving the robot towards the goal. This technique attempts to balance the two, by measuring the entropy of the probabilistic distribution of the robot's belief states, and performing actions to reduce this entropy to a certain level before moving towards its goal. In other words, figure out where the robot currently is, and once known to a certain degree of probability move toward the goal until the entropy increases to a certain point, where the robot relocates itself. This is quite similar to what is trying to be accomplished in the work presented in this thesis, which is the balance between coverage of a scene, and the accuracy of representation.

Testing all the heuristic algorithms under several different simulated and real robotic platforms, the authors noted that the second method, termed the most likely state, yielded the most consistent results, and very rarely failed.

Pineau *et al.* [53] are able to make the POMDP process tractable by using a point based value iteration algorithm (PBVI). The PBVI technique selects a small number of belief points, and updates their values, as well as the value gradients, which cuts down on the number of computations needed from an exponential complexity to a quadratic complexity. The small set of belief states selected by the PBVI algorithm are the ones that are most probabilistically relevant for the task, and continuously update the selected tracked belief

state set through value backups. The value backup operation is traditionally the determination of the value function at the next instance in time, based on the current and previous value functions. What this means is that when using the PBVI algorithm, instead of keeping track of each possible value state along the path, it updates the current value, and adds the appropriate new belief states to track the values until the goal is reached. In a mathematical sense, this yields a projection operator instead of the expansion of the value state space, thereby maintaining a 1:1 correspondence between values and tracked belief states, as opposed to a set of values for each belief state, which increases in dimensionality over time.

Finally, neural programming can be used to approximate MDP and POMDP performances [39]. Haykin discusses a couple different ways that neural networks can aid in a decision process. The first is to approximate the iteration process that the MDP/POMDP uses to find the optimal actions, or policy, through the use of simulations using a neural network. The second is to perform this process online by using the parameter space of the problem in a process called Q-Learning, which is comparable in action to the value iteration method of solving MDPs. The assumption made in Q-Learning is that the states are fully observable (i.e. MDP only). Either method results in determining the suboptimal result more efficiently than traditional methods. Haykin [39] and Fu [41] discuss winner-take-all neural networks which can be used in classifiers, identification problems, and other similar decision systems.

What all of the methods presented in this section have in common is that they make a decision based upon an action-reward based system, which attempts to optimize the task, based upon the current state, as well as the reward for performing a particular action. They vary in terms of the assumptions they make, and how they make the problem computationally tractable for real-time decisions. In the scenario presented in this thesis, the current state is the current range points acquired, the action is the next point to acquire, and the reward is the amount of improvement which results from that acquisition.

2.4 3D Acquisition, Representation, and Processing Techniques

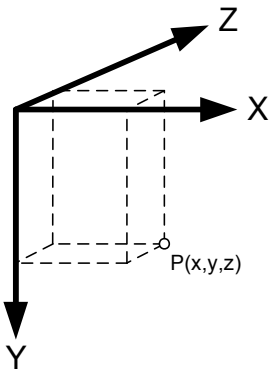
The work presented in this thesis involves minimizing the data collected for 3D imaging. Therefore the manner of how 3D data is acquired, as well as its representation and usage, is considered in this section. The first subsection describes various coordinate systems in

which to represent 3D data. The second subsection discusses methods of acquiring 3D data, using different sensing modalities. The third and fourth subsections discuss the two most popular methods to represent 3D data, which are a surface mesh and a voxel grid. Finally methods which process these 3D models are presented in the final subsections.

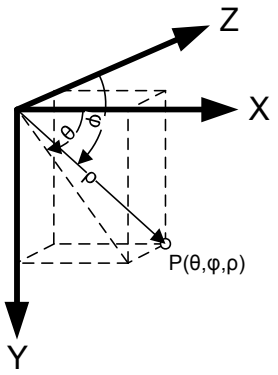
2.4.1 Coordinate Systems

There are several different coordinate systems that can be used to represent 3D data. For the context of this thesis, assume that all formula can be represented within an arbitrary coordinate system, (c_0, c_1, c_2), where it can be thought of that c_0 and c_1 are the independent variables along which the sensor measures the value c_2, i.e. $c_2 = f(c_0, c_1)$. c_0 can be presumed to be the horizontal measure (angular or displacement), while c_1 can be presumed to be a vertical measure (angular or displacement). Furthermore, when other coordinate systems are used in the development, it is done so to better conform with the literature, or to insure clarity of development.

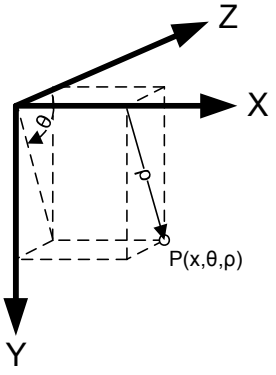
The most familiar coordinate system is probably the Cartesian coordinate system as displayed in Figure 2.4 a), which is formed by the three mutually perpendicular displacement axes, $(c_0, c_1, c_2) = (x, y, z)$, whose values may be any arbitrary value. Spherical coordinate systems, as illustrated in Figure 2.4 b), contain azimuth, elevation, and radial components $(c_0, c_1, c_2) = (\theta, \varphi, \rho)$. The azimuth typically ranges between -/+ 180 degrees, the elevation between -/+ 90 degrees, with the radial component being larger than zero distance units. Cylindrical coordinate systems, as shown in Figure 2.4 c), contain a Cartesian displacement component, an azimuth component, and a radial component which is perpendicular to the Cartesian displacement component, $(c_0, c_1, c_2) = (\mathrm{x}, \theta, \rho)$. The Cartesian displacement component may have any arbitrary value. The azimuth component must have a value between -/+180 degrees, while the radial component is a distance that is larger than or equal to zero. Finally, this thesis makes use of an angular projection coordinate system, as displayed in Figure 2.4 d), with two independent angular components and a distance component, $(c_0, c_1, c_2) = (\theta, \varphi, \rho)$, for use with sensors which are projective based. Appendix A gives a more detailed development as well as the mathematical transformations between the different coordinate spaces.



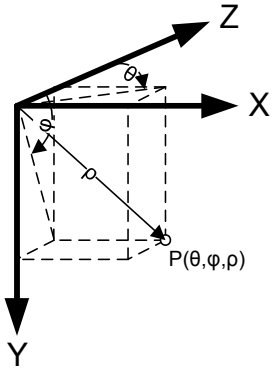
a) Cartesian coordinates



b) Spherical coordinates



c) Cylindrical coordinates

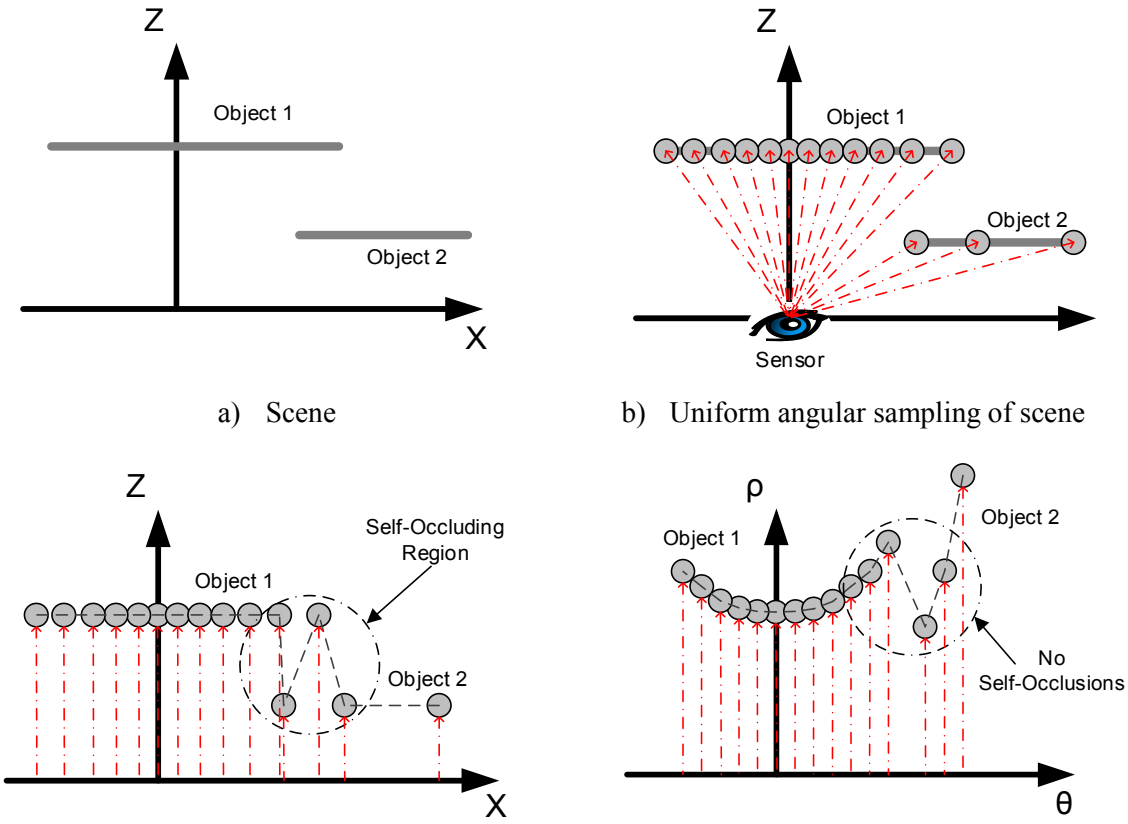


d) Angular projection coordinates

Figure 2.4 - Illustrating different coordinate systems in relation to Cartesian coordinates

The reason why these coordinate systems matter in the context of this thesis, is that by transforming range data from the native coordinate system employed by the sensor, to another one (such as the usual Cartesian coordinate system), perceived artefacts can be introduced in the scene. This thesis uses the term 'self-occlusion' to describe these perceived artefacts, and Figure 2.5 illustrates how these artefacts are formed. Figure 2.5 a) illustrates a generic scene which is then sampled in Figure 2.5 b) by a sensor at the origin which acquires range data by sampling uniformly about a rotation axis, as illustrated by the dashed red arrows. Figure 2.5 c) shows the same sampled data plotted in Cartesian coordinates. An observer who perceives this data in a Cartesian manner, views the ranged data points as the dashed red arrows shown in Figure 2.5 c). Notice that within the dashed circled region, that one of the observed points from object 1 should be occluded by object 2, but the observer is able to perceive it anyways because of the way the data was natively acquired. Also notice that there are a pair of points that almost share the same viewing vector which would put the

point on object 1 in occlusion of the point on object 2. These are the particular artefacts that are termed self-occlusions within this thesis. Furthermore, Figure 2.5 d) shows the same data plotted on an angle/depth coordinate system, which is the native coordinate system on the sensor, where the observer perceives the points as shown by the red dashed arrows spaced uniformly along the θ axes. It evidences that for the same region there are no self-occlusions present in the plotted data. Self-occlusions become an issue if present, since they confuse the representation of the objects in the scene, as it becomes difficult, if not impossible, to determine if a particular location that the sensor can acquire belongs to the same object as the rest of the local neighbourhood, and indeed tosses into doubt the assumption that measure points can be used to estimate other locations within their local neighbourhoods.



a) Scene

b) Uniform angular sampling of scene

c) Sampled points plotted using Cartesian coordinates

d) Sampled points plotted used native angle/depth Representation

Figure 2.5 - Demonstration of Cartesian self-occlusion present in data sampled angularly

2.4.2 3D Data Acquisition

The most common categories of acquisition modalities for 3D datasets are time-of-flight and triangulation based sensors. Time-of-flight sensors are active sensors, meaning that they emit energy, which include several different sensor technologies such as sonar, radar, lidar, and some laser range finders. In all the cases, the sensor consists of an emitter, which emits a signal, which then travels until it encounters a reflective object, from whence it reflects back to the sensor, where it is recorded and the time between emission and reception is determined, as illustrated in Figure 2.6. The depth measurement is estimated as a function of time. In the case of a single point time-of-flight laser range finder, the depth would be half of the time divided by the speed of light (since the distance corresponds to half of the time it takes for the laser to be emitted, travel to the object, reflect back and then be recorded). Sonar (pressure waves), radar (electromagnetic waves), and lidar (light waves) sensors are similar, but emit a pulse that is not a single point, and so multiple reflections, directions, and times must be recorded to reconstruct a scene.

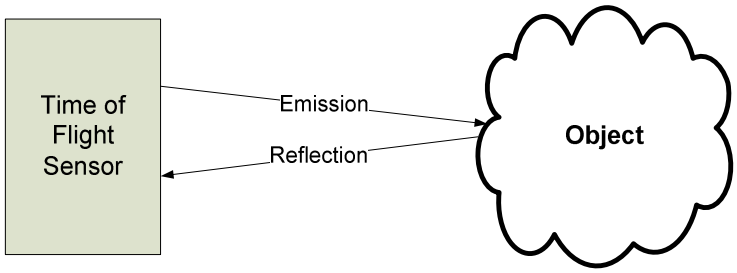


Figure 2.6 - Simple diagram of a time of flight sensor

Triangulation based techniques include stereo vision, structured light, and shape from motion [57]. Stereo vision is a passive technique which performs the matching of features between two or more different images acquired from different cameras. Structured light is an active based approach, where a known pattern is projected upon a scene, which is then captured by a camera. A matching of features between the projected pattern, and the acquired image is then made. In either case, the features which are matched correspond to a known point in space, but since only the 2D projection of the point is known, the position in space cannot be calculated from only one image, but instead a pair of rays are used. As Figure 2.7 illustrates, these rays pass through the focal points of the respective camera or projector and pass through the corresponding feature point on the image plane. The object is located where

these rays intersect (in the ideal case), and by measuring the displacement in 3D space between the cameras and projectors, called the baseline, the 3D coordinates of the object can be determined by using the triangle formed by the two rays and the baseline between cameras and projectors. Shape from motion operates in a similar fashion, except there is only one camera, and the object is acquired at two instances in time (see Figure 2.8). The acquired features must also include some measure of size in addition to position, such that relative size difference and the baseline between features can be determined. Unless the size of the object is known *a priori*, shape from motion can only determine the 3D position up to a scale factor.

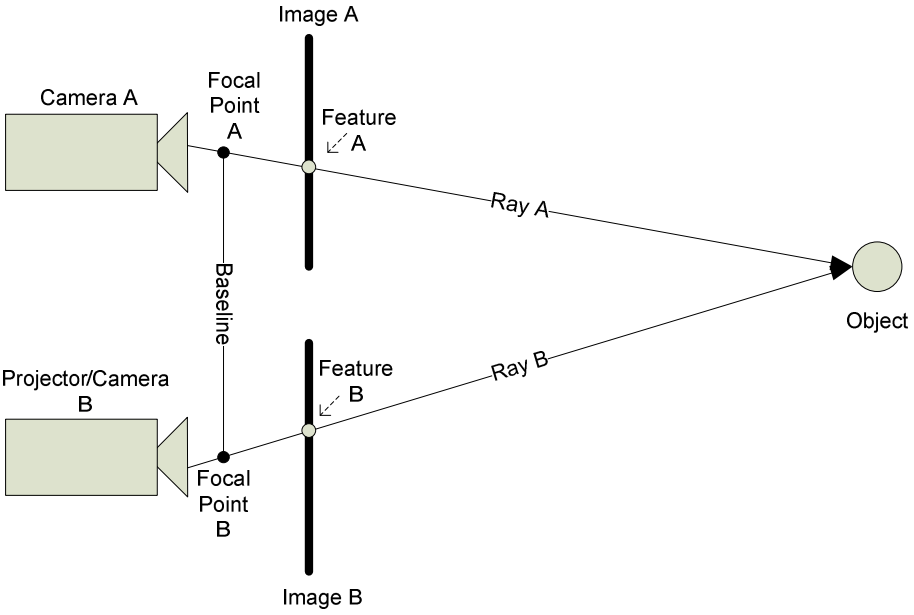


Figure 2.7 - Triangulation of a structured light/stereo vision system

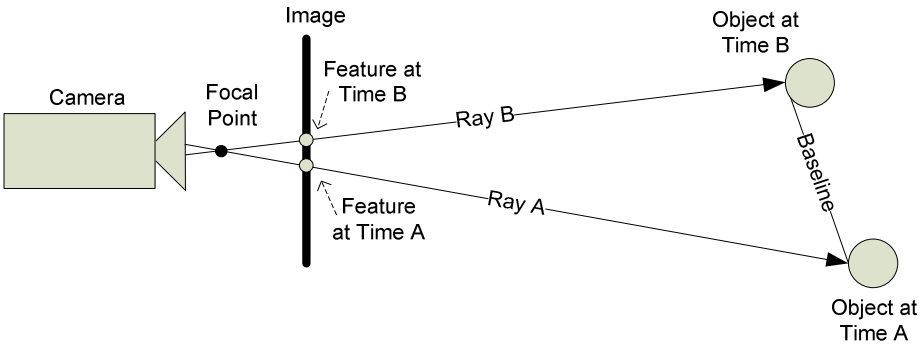


Figure 2.8 - Triangulation of shape from motion

The commonality between all the sensor types listed above is that they emit their energy and receive reading by rays that propagate through a single point, and as a result can be natively used in angular or projection based systems, such as the spherical coordinate, cylindrical coordinate, and angular projection coordinate systems introduced in Section 2.4.1 and detailed in Appendix A.

2.4.3 Surface Mesh

Once a 3D image has been acquired, it is generally useful to represent it in a format that allows a user to view and manipulate it. One such representation is the surface mesh. A surface mesh is a series of points which have a surface fitted to them. The most common shape used to link the points to create a surface is the triangle, and the points may have other properties associated with them other than location, such as surface normal at that location, colour, texture, and material. This section discusses methods for creating a 3D surface mesh.

Delaunay Triangulation [58,59] is a method of creating a convex triangular mesh such that for all triangles in the connected mesh, there will not be a point that lies within the circle defined by the vertices of another triangle. Figure 2.9 shows two scenarios with the same point configurations, but with two different triangulations. On the left is a proper Delaunay triangulation, where the circle defined by triangle ABC (in blue) does not contain point D, and the circle defined by the triangle ACD (in red) does not contain point B. On the right is a non-Delaunay triangulation, where the circle defined by triangle ABD (in blue) contains point C, and the circle defined by triangle BCD (in red) contains point A. As can be shown in the figure, to change the triangulation on the right of Figure 2.9 to be a Delaunay triangulation, what is known as an edge flip operation can be performed, which is simply removing edge BD and adding in edge AC. Delaunay triangulations are popular since they tend to avoid long skinny triangles, thereby keeping the vertices of the triangle close to each other (i.e. more local).

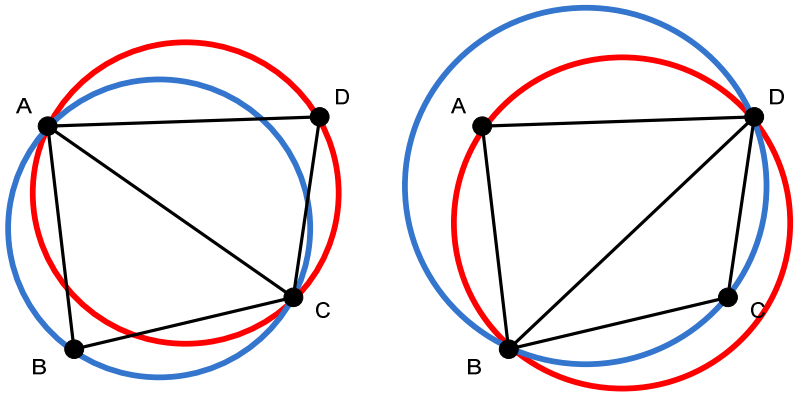


Figure 2.9 - Proper Delaunay triangulation (left) and non-Delaunay triangulation (right)

There are many methods to construct a Delaunay triangulation, from divide and conquer algorithms that scale well on parallel hardware, to incremental algorithms that allow the addition of one point at a time. But they all follow the same principle of doing what is called an in-circle test, which tests to see if a point lies within the circle defined by a triangle, followed by an edge flip if the point does lie within the circle. This series of operations continues until all triangles meet the criteria of a Delaunay triangulation. Additional extensions and modifications to the Delaunay triangulation algorithm exist, which allow for constraints along edges, as well as mesh boundaries, and into multiple dimensions. The constrained algorithms either allow non-Delaunay triangles to exist close to constrained edges, or add in additional points along edges such that all triangles meet the Delaunay criteria. Multiple dimension Delaunay triangulation algorithms are just extensions, where circles become spheres and hyper-spheres, and triangles become tetrahedra and hyper-tetrahedra. But for 3D surface mesh generation the projection of the 3D points onto a 2D surface is performed, and then a 2D Delaunay triangulation of the visible surface points is usually performed since it requires many less computations than a full 3D Delaunay triangulation.

Mederos *et al.*'s technique [60] for constructing a surface mesh from a point cloud is a multi-step process. The first step is to split the point cloud into a series of clusters which attempt to maintain a similar curvature within the geometric neighbourhood. If the curvature does not appear to be similar, the volume is subdivided into smaller ones until regions contain points that meet the curvature requirements. The curvature requirements are determined by using the eigenvalues of the covariance matrix from the points in that particular sub volume,

and if the eigenvalues are all of similar values, then the surface is mostly flat, and hence the points share a similar curvature. Once the point clustering has been performed, the second step is to reduce the number of points using a moving least squares approach, such that a collection of points can be represented by a single point. The third step is to build the triangular surface mesh incrementally that meets the requirements of a restricted Delaunay triangulation. The final step is the refinement of the final mesh by performing tessellation of the 3D surface mesh, by adding points on edges that are longer than a certain threshold. While the final result does not contain many points from the initial point cloud, it does represent the point cloud in a moving least squares approximation of the surface, and effectively performs filtering of the input points.

Wang *et al.*'s method [61] constructs a mesh from a 3D point cloud of both manifold and non-manifold surfaces relying solely on the point cloud data. The technique is a 3-step process. The first step is to create a voxel representation of the point cloud, and then the gaps between voxels are filled. In order to determine whether or not a voxel is to be classified as a gap relies on determining the labelling of existing voxels as a boundary voxel, a junction voxel, a surface voxel, or a half surface voxel. If the neighbouring voxels are correctly labelled, and the spacing is within a certain threshold, then the voxels in between labelled voxels are labelled as gap voxels, otherwise they are labelled as empty space. After the gaps are filled in, extra voxels are removed such that the surface is only one voxel thick. Finally a surface that goes through the voxels is determined by fitting a surface patch within each set of connected boundary voxels following the curvature defined by the contained surface voxels. All of these individual patches are merged together to create the final mesh, with the points of the mesh corresponding to a location in the voxels (usually centroid). This technique does not rely on the geometry of the points, but it unfortunately relies on a fine regular sampling density, otherwise the resulting voxel space will be too thin to generate a reliable mesh.

Rusu *et al.*'s technique [62] to create a surface model based on 3D point clouds relies on point cloud segmentation and then fitting geometric primitives to those point clouds. The algorithm, since it is designed for robotic manipulation on flat surfaces, makes certain assumptions of any acquisition, namely that there is a flat surface which is parallel to the

world z-axis (up/down), and relies on this assumption to aid in segmentation. The first step in the algorithm is to generate surface normals for each point, either through normal data provided by the sensor itself, or estimated by the geometry formed through its nearest neighbours. Any point with a normal which is parallel to the world z-axis is labelled as possibly belonging to the work surface. Each cluster of these points has a planar model fit to it, and a bounding polygon is computed. All clusters of points within the polygon bounded work surface are extracted as separate items, and geometric primitives are then fitted to these clusters, in order to produce a list of shapes, as well as a surface model of the work environment. The technique described works well within controlled environments, is effective at dealing with noisy data (due to the use of geometric primitives, and irregularly sampled data), and inherently segments objects at the same time as mesh generation. But unfortunately it will not work well in more general complex scenes where a planar surface or background cannot be expected to be present.

Schall *et al.* [63] describe a process on how to filter noisy point cloud data, on which a surface mesh is then applied using Delaunay triangulation. The authors state that Delaunay triangulation does not work well on raw point clouds due to the effect of a single outlier point causing an extruding sharp point, or points, from the desired surface model of the cloud, and hence prefiltering of the results is highly desired to remove the offending points. First in a local cluster of points, a planar surface is determined that produces an optimal approximation of the local surface. This is accomplished by iteratively determining the plane of best fit using a likelihood function, which effectively adjusts the weight, or likelihood that a point belongs to that surface, of each point at each iteration. After convergence of the likelihoods and the planar surface, a gradient-ascent maximization process from which all points are moved to positions of higher probability is applied, which effectively smoothes the surface while dramatically reducing outliers. This technique appears to perform well at filtering noisy 3D data, and could be adjusted to provide an initial surface mesh if the fitted planes were used to construct the mesh. The sole drawback, which is the same as any filtering application, is the fact that small scale features will inevitably be filtered out, and fine grained texture can be lost. This technique could be modified to be an effective outlier detection system by just removing low likelihood points from the cloud all together.

2.4.4 Voxel Grid Representations

Another common representation for 3D data is the voxel grid. A voxel is a 3D discrete volume unit analogous to a pixel from 2D image processing, and can have several properties attributed to it, such as colour, occupancy, and material. There are several different types of voxel grid representations. There is the naïve representation where all voxels on the grid are of identical dimensions, and then there is the multi-resolution octree representation, where voxels are represented as nodes of an inverted tree structure. At the highest most level of the octree, the entire space is represented by a single large voxel, and as one descends down through the octree, the voxels become smaller in a progressive manner, typically being recursively subdivided by two along each spatial dimension. This means that at a particular octree level, all the voxels will be the same size, and that at level n-1, the voxels will be 8 times larger in volumes, and at level n+1 the voxels will be 1 eighth of the volume. A naïve representation provides intuitive direct access to every voxel in space, and is a natural extension from working with pixels in images into the third dimension, while a multi-resolution octree allows for the compression of like information, such that regions where there is no change in contents can be represented by larger voxels, and regions of large variation can be represented by many smaller voxels. Different types of data can be stored in voxels, just like with images. The most common element stored in voxels is occupancy of the voxel, i.e. does something other than air exist in that unit of space. This section describes methods for creating and managing voxel grids.

Connolly [64] discusses how to build up a 3 state octree from a range data acquisition stored in a quadtree (a 2D octree). The three states are defined as 'seen but empty', 'occupied', and 'unseen'. The technique as described first creates the octree with its nodes labelled as unseen. To add information from the quadtree to the octree, each quadtree node is projected through the octree from the octree reference frame using the ray tracing of the corners of the quadtree. New octree nodes are created to be at the same resolution as that of the quadtree data. When carving away at the octree using the projections, all nodes until the depth measurement stored in the quadtree are labelled as 'seen but empty', the nodes where the depth measurement occurs are labelled as 'occupied', as any cells behind. The author finally provides criteria to avoid edge cases and pockets of unseen voxels in what should be the visible area.

Elfes [65,66] describes a technique for using occupancy grids for robotic navigation. An occupancy grid models free space by the probability that the particular sub volume is occupied or not, with a probability of 1 corresponding to absolute surety that the sub volume is occupied, 0 corresponding to the absolute surety that the sub volume is unoccupied, and 0.5 being uncertainty of whether or not there is occupancy. Elfes technique starts by characterizing the stochastic model of the sensor so that the probability function of its acquisitions can be applied to each measurement made by the sensor. This occurs because all data contains noise and uncertainty, especially when the sensor is mounted on a moving mobile robot platform. Occupancy grids are an invaluable tool for unstructured, changing, or unknown environments, and allow a system to reacquire data in the same region of space until enough readings provide confidence that the space is actually occupied or empty. Bayes theorem is used when updating cells with new information, and thresholding of probability levels can provide more discrete and conventional labelling of empty, occupied, and unknown. The paper [65] provides the equations for a sensor based on Gaussian error model. It then goes on to provide some implementation results on a live robotic platform using sonar. The benefits of an occupancy grid representation is that it inherently handles uncertainty of measurements, but it unfortunately emphasizes accuracy over coverage as certainty in measurements can only occur over several acquisitions of a particular point.

Payeur *et al.* [67] apply probabilistic occupancy grids using octrees for modelling 3D environments. The main issues that were discovered when applying 2D occupancy grids to 3D were that the computational cost increased dramatically. To solve this issue, the authors develop a closed form approximation which they term the occupancy probability distribution function (OPDF), as well as modelling the grid within an octree, such that voxels of similar occupancy probabilistic are merged together. As with Elfes [65,66] solution, the first step is to obtain a probabilistic model of the sensor, from which an approximation of the function that Elfes used is applied to each cell using Bayes rule. Building the octree over several scans is accomplished similar to what Connolly describes in [64], except a spherical octree is used to store the initial probabilistic occupancy of the sensor acquisition, and the spherical sensor occupancy grid is merged with the global Cartesian occupancy grid using Bayes rule to create the model of the scanned environment. By using an approximation of the function, as well as establishing limits to the range of influence of the measurements, the modelling

process was sped up dramatically. The authors obtain good results modelling high voltage lines using a laser scanner.

2.4.5 3D Feature Extraction Techniques

After a 3D image is acquired, if a computer vision system wishes to perform a task based upon the data, the latter must be processed. Typical processing techniques in 3D computer vision, as in 2D methods, rely on processing the image to locate features, so that objects and regions of interest can be extracted from the data. This section discusses some of the techniques present in literature on this subject.

Pauly *et al.* [68] propose a technique for feature extraction from a surface point cloud. The feature type that the authors have selected to use are lines which represent edges between significant changes in curvature. First, the authors use a guided k-nearest neighbours technique, from whose points are used to determine local curvature via eigenvalues of the point covariance matrix. To determine how dominant a feature is, they analyse it at different scales. The authors take advantage of the fact that the process of scaling/subsampling in 1D and 2D is effectively the same as using a low-pass filter, which is similar to what happens when the local neighbourhood is made larger - high frequency changes are effectively suppressed, while the low frequency areas stay present (i.e. 3D surface smoothing). Each point at each scale is given a weight, which is proportional to the variation of the eigenvalues of the point covariance matrix, as to whether or not the point belongs to a feature. After performing a threshold of the weights, by either an automatic method related to the statistics of the weights or by human intervention, points belonging to features are extracted, and 3D snakes are fitted to local neighbourhoods of feature points, which are then considered the detected features. The operator of the algorithm also has a choice on what scale of features to select, from large scale features, or small scale features, or the scale of the features can be determined algorithmically by looking for a maximum weight, or other statistics, across scales, and selecting that scale to extract features from. The technique described relies solely on the point data, and can effectively determine large scale features, as well as finer detailed features, and as the algorithm relies on a more or less regularly sampled dense point cloud in order to prevent biasing due to sample clustering, this technique is unsuited for selective sensing which does not guarantee a constant sampling density throughout the acquisition.

Woo *et al.* [69] have created a segmentation algorithm that works on 3D surface point cloud data for the purposes of creating CAD models. The first step in the process is to obtain the point normals. If the 3D sensor does not provide them directly, a meshing strategy such as Delaunay or another algorithm is to be performed, from which an estimate of the point normal can be calculated. After the point normals are determined, the points are put into a multi-resolution octree, and subdivision of the octree's cells occurs when the standard deviation of the point normals exceeds a predefined tolerance. Since smaller octree cells are inherently formed when the variation is large, these correspond to transition regions of the object. To perform the segmentation, the smaller cells are labelled as edges and effectively removed from the octree. What remains is considered different segments. To label connected cells as belonging to the same segment, a seed growing algorithm is used. Once the regions have been labelled, a surface is fit to the points of the region. Due to the use of the multi-resolution octree, this algorithm relies on data that is regularly sampled with a fairly high density, otherwise holes will exist in the octree, and what should be labelled as a single segment will have discontinuities within, and as result is not suitable to selective sensing, as selective sensing produces data which is not guaranteed to be regularly sampled.

Yogeswaran and Payeur [70,71] build upon the technique put forth by Woo *et al.* [69] by using a variable threshold that is automatically adjusted upwards in a linear fashion as the octree is transversed from the root node to the smallest leaf. This prevents smooth curving surfaces and low level noise from erroneously being detected as features, while at the same time reducing overall tree depth. Taking advantage of the fact that leaf nodes will contain variation, and hence are indicative of features, all other triangles that are not contained by these lowest level leaf nodes are classified as non-feature triangles, with the other groupings of triangles being classified as potential features. An analysis of relative occupancy is made, based on the remaining feature triangles and adjacent octree nodes are considered to belong to the same feature or edge if they share similar occupancy and surface normal variations. As opposed to Woo *et al.* [69], Yogeswaran and Payeur do not remove these edge triangles, but they are instead left in the model to determine feature connectivity. The surface normal variation and occupancy levels of these connected regions are then used to classify features. Since Yogeswaran and Payeur in [70,71] are interested in defect detection, the classification of features is based on intentional features (features that are estimated as being part of the

object, and tend to be larger scale and smooth), and defects (that tend to be abrupt and inconsistent with the surface). Due to the adjusting of the thresholds during the creation of the octree, this method is not as susceptible to variations in sampling density, but is not totally suitable for selective sensing as the focus only on certain types of features is made, as opposed to capturing all features.

Hubeli and Gross' method [72] for feature extraction works directly on meshes. A feature in this context is defined as a piecewise collection of edges belonging to the mesh. The first step of their algorithm is to assign a weight to each edge in the mesh, which corresponds to the probability that the edge is a feature. There are several methods that the authors explore to perform the operation, depending on the scenarios of noise and mesh density, with one set of methods calculating the weights based on the angles of adjacent surface normals, or opposite point normals, and the other set based upon fitting a polynomial through the edges, and then using the second order derivatives to determine where transitions occur. After the edges are assigned the appropriate weight, they are thresheld with a hysteresis function to provide a set of connected edges that belong to a particular feature. To eliminate false positive features, if a feature only contains a few edges, it is dropped and considered noise in the meshing process. The remaining feature edges then have patches applied to them so that edges become connected surfaces, and a skeletonizing algorithm prunes unnecessary edges from the patches. Finally each fully connected feature is given an importance rating, which is a combination of edge lengths and weights that belong to the feature, such that unimportant or small scale features can be eliminated. The authors also extend their technique to be multi-resolution through vertex clustering algorithms, so that dominant features can be determined faster, with more precision. If desired, points around that feature can also be re-added so that the feature can be refined as well as determine any smaller scale feature which may be desired. From the published results this method looks to be fairly robust and is able to effectively determine large scale transition areas, as well as deal with data with small amounts of noise and varying levels of acquisition density, but the authors did not investigate methods to automatically tune thresholds and is wholly reliant on operator tuning for effective performance.

Lee *et al.* [73] demonstrate a 3D segmentation of scanned point data for the purposes of reverse engineering and generating CAD models. The first stage is to generate a mesh for the points using the Delaunay algorithm, or another technique. After the mesh is generated, triangle surface normals are calculated, and neighbouring triangles are merged into regions if their surface normals are similar. When the merging process is complete, boundary triangles of the mesh are found which corresponds to regions where the triangle surface normal varies by a predefined amount. After boundary triangles are determined an edge is classified as gradual or sharp depending on how much the surface normal varies over the region. A region with a gradual variation of the surface normal is deemed as a rounded area, while a sharp variation is determined to be a sharp edge. The difference between the two is a function of surface area of the transition region, as well as the angular difference over that area. By categorizing between these two different types of edges, the final models can be refined by providing a more accurate rendition of the edge locations, instead of having the edge being jagged or smooth. This algorithm can work with a varying sampling density, since it takes into account the triangle areas, as well as the direction of the normals, and can provide more accurate location of edges even if they are not directly captured by the sensor, which is of interest for selective sensing in highlighting regions to acquire with more points.

Kim and Tsiotras [74] describe a technique for segmenting multi-resolution quadtrees and octrees by using the center of the cells as the value in the tree structures. The multi-resolution nature of the trees allows for a compact representation of the geometry of space, effectively bundling a homogeneous area into a single large cell, and areas with greater variation into many smaller cells. Unfortunately by using the cell centers, as opposed to the cell corners, interpolating the data between two cells is complicated, since one has to navigate through the tree structure, and cells are not necessarily stored at the same resolution. The solution implemented in the technique involves restricting the tree connectivity in order to improve computational efficiency by implementing the interpolation weights in a lookup table. The restriction that the authors impose is that the tree depth of neighbours must be only one resolution difference between adjacent cells. This allows the lookup table for a quadtree to be only 15 entries, and the octree to be only 255 entries. These weights are the bilinear/trilinear interpolation weights for the connectivity scenarios described, which allows the quick calculation of interpolation values within the quadtree/octree. With the

interpolation problem solved, a partial differential equation based segmenter can be used. The authors' show that a dramatic speed increase is achieved using their technique against more traditional brute force techniques. As this technique assumes a uniform sampled surface, it is not suitable for use in an incremental selective sensing algorithm.

In an attempt conducted at an early phase of investigation towards this thesis, Curtis and Payeur [75] develop a method to segment a surface point cloud for the purposes of determining where more data should be acquired to better represent the scene in a selective sensing application. The technique first relies on obtaining a course acquisition of the scene. A Delaunay triangulation is then applied to the point cloud, followed by the determination of the normals of each triangle in the mesh. The surface normals are then used to estimate the point normals, with the weighting factor being the inverse of the triangle area. The inverse of the triangle area is selected since densely sampled regions are represented by smaller triangles and are more local than sparsely sampled regions represented by larger triangles, thereby giving more weight to regions where the density, and hence locality is greater. After the point normals are determined, they are inserted into a multi-resolution octree, which is subdivided based upon the standard deviation of the point normals contained in each voxel. The voxels are then combined into segments based upon neighbouring voxels determined from the connectivity of the original Delaunay triangulation, and depending on how much deviation there is between the normals of the segments to be merged. This method worked well at segmenting 3D datasets of non-homogeneous density, but ultimately proved unsatisfactory at providing a framework to aid in identifying regions which require more points to properly represent the scene due to a combination of complexity of the implementation and lack of adaptability to use in a selection process, and as a result is poor for the task of selective sensing.

2.4.6 Compression

Compression is a process which reduces the size of a dataset. Compression can be divided into two categories, lossy and lossless compression. Lossy compression reduces the size of the dataset typically by reducing the accuracy and the precision of the data, thereby making it impossible to perform decompression to obtain the original dataset. For example, the lossy JPEG algorithm [76] for compressing image data, first subdivides the image into a set of 8x8

sub-images. These sub-images are then converted using the discrete cosine transform (DCT), where lower frequency components are encoded with higher precision (more bits) than higher frequency components. After this quantization, a type of entropic encoding called run length encoding (RLE) is used to further compress the data from each sub-image. On the other hand, lossless compression reduces the size of the dataset by relying on the entropic qualities of the data involved, and allows for the original dataset to be completely and precisely reconstructed. An example of this is Huffman encoding [77] where a dictionary of symbols, and their corresponding probabilities, are constructed from the source data. Symbols which have a higher probability of occurrence are assigned a shorter code word to represent them, while lower probability symbols have longer code words. While the mean symbol length is approximately equal to the mean code word length, the fact that the higher frequency symbols (and therefore higher probability) are represented with short code words, compression of the dataset can be achieved.

Compression is measured by several metrics. Data reduction achieved is typically represented as a compression ratio or bit rate, but may also be represented as data savings ratio. For lossy compression however, as information is dropped the dataset effectively loses quality, and as a result a quality measure is used to gauge this effect, and it is used in tandem with a data reduction metric in order to provide an overall measure of compression performance. There are several quality measures used, including the peak signal-to-noise ratio (PSNR), mean squared error (MSE), mean absolute error (MAE), and subjective perceptual analysis [76]. In the literature, PSNR is the most common metric used quality metric due to its simplicity, and it can be calculated from the MSE, shown in eq. (2.22), by the use of eq. (2.23), where $I_1(i,j)$ and $I_2(i,j)$ correspond to the i^{th} row of m rows, and j^{th} column of n columns of the images I_1 and I_2 respectively, and MAX is the maximum quantization range of the image. The drawback to using the PSNR and other mean error measures, is that it deals with only differences in pixel value, and does not necessarily capture the perceptibility of the error nor change of location errors.

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left(I_1(i,j) - I_2(i,j) \right)^2 \tag{2.22}$$

$$PSNR = 10 \cdot \log_{10} \frac{MAX^2}{MSE} \tag{2.23}$$

Compression of 3D representations vary according to their representation. The compression of 3D meshes can be accomplished in many different ways [78]. A 3D mesh consists of both vertices and the edges which interconnect them. First, the mesh can be resampled, which causes the reduction of the number vertices and edges needed to encode the surface, while maintaining the overall features of the surface. Secondly, the precision of representation can be changed. Both of these lead to lossy compression of the mesh. Thirdly, entropy encoding can be used, which will allow the lossless compression of the input mesh, or in the case where entropy encoding is applied to the first two options, will provide an even greater overall compression. As in 2D image compression, the mesh can be transformed into a different representation domain. One of the more common transformations is the transformation of a surface to a point, and vice-versa, similar to the representative duality of the Voronoi diagram and the Delaunay triangulation. When using lossy compression on 3D meshes, it is imperative that the boundaries of the surface do not change, only the texture of the surface details should be smoothed, in order to retain the visual quality of the mesh.

Compression of depth maps is similar to the compression of 2D images, except that with depth maps, instead of each pixel representing a colour, it represents a depth. Furthermore, as the quality of a depth image is highly dependent on the location of its edges that are located in the high frequency regions in the transformation domains used for quantization, traditional lossy 2D algorithms will blur the edge into a slope, which in many situations is not desirable [79,80]. In order to preserve the locations of these hard edges several techniques have been developed. Krishnamurthy *et al.* propose using region of interest (ROI) coding from the JPEG-2000 standard to ensure that edges determined using a classical edge detector are preserved, while the rest of the depth map is compressed as it would regularly. Furthermore, they reshape the dynamic range of the depth maps, such that closer distances are compressed, and further distances are expanded. The combination of these two methods yielded an increase in the PSNR at the same bit rate. Sarkis and Diepold [80] approach the problem by subsampling the frequency domain representation using a specially formed random subsampling matrix. In order to preserve the discontinuities, total variation

constraint is used in conjunction with a non-linear conjugate gradient minimization technique in order to determine the coefficients of the compressed representation.

2.5 Chapter Summary

The current intelligent selective sensing techniques rely on an ad-hoc stopping of neural network training [6,7], or techniques that are purely deterministic [8]. Other techniques for optimal scanning have focused on evaluating individual patterns for specific applications [9], or switching between multiple sub-optimal patterns [16]. A few techniques take into account the estimate value of missing data [13,14], as well as optimizing the resulting 3D output [15], but they are for different application domains. On the other hand NBV algorithms perform well at globally optimizing the acquisition of a scene from many different viewpoints, but that is a different problem than that being investigated in this thesis, although they provide some interesting insights into the problem, in particular that of MacKinnon [28,29,30], who measures several quality metrics and focuses on viewpoints to optimize quality. MacKinnon's work is developed in the context of a specific sensor and is applicable towards solving the multiple points of view problem, which contrasts with the objective of this thesis to design a technique that is applicable to many different sensors and encompasses optimizing the acquisition of a single point of view.

In order to have a process to decide where the optimal location is to acquire a new data point from within a 3D scene, estimates of the value of the currently unknown point depths are needed, as well as a measure of the error on those estimates. With that goal in mind, there are several estimation techniques which perform well in different situations. General approximation theory allows the determination of whether an approximation is sufficient for a given scenario, and allows for the use of a simple system to be used instead of a more complex system, thereby reducing the computational burden imposed by the system without the addition of much systematic error. Functional estimators, splines, Ordinary Kriging, and neural networks are all good techniques for interpolating within a dataset. Of all these methods Ordinary Kriging, paired with an appropriate semivariogram model, allows the estimation of both a mean value, which in this application is depth, as well as its variance, implicitly within the formulation of the technique. Due to Kriging supplying both metrics, it

is a good fit for applications where knowledge of error is desired, such as selective sensing as is being presented within this thesis.

Once an estimate of the depth of the possible, but currently unknown, sampling locations is performed, a decision of where to scan next must be made. The MDP/POMDP are optimal decision processes, but the mathematics are unwieldy and inefficient, and as such are not suitable for the purpose of selective sensing as presented in this thesis. Ad-hoc methods, on the other hand, are generally used to try to capture the concept of action-reward that MDP/POMDP's use, but at a much lower computational cost, which is of definite interest to the scope of this thesis. Finally, the use of neural networks, through neural programming, can be used for these ad-hoc approximations of an optimal decision process, at the cost of the complexity and time of training the network. As the goal of the selective sensing technique is the ability to deal with arbitrary scenes, developing a training set is counterproductive.

In order to adequately perform these tasks, knowledge of how data is acquired, manipulated, and represented is required. In order to avoid self-occlusion in the dataset, and so that the depth may be represented as a map, or an image, mapping the data to the sensor's native coordinate system is desired. A depth map allows the trivial application of the estimation techniques to be made, but a depth map by itself is not a complete 3D representation. More common 3D representations for performing 3D processing are the surface mesh and the volumetric voxel grid. A surface mesh allows gaps in the data to be approximated by a polygonal patch, where a probabilistic occupancy grid does not provide any assumption on those values. On the other hand, a probabilistic occupancy grid only provides a higher accuracy measure of where a surface is actually located given a high enough resolution, where a surface mesh contains no inherent discretization and can provide as high accuracy measure of the surface location as the sensor can provide. Due to the complementary duality of these two representations, namely that surface mesh representation provides a better measure of accuracy and a probabilistic occupancy grid provides a better measure of coverage, both will be used in combination to determine the next point to acquire.

Chapter 3. Improvement Metrics and Selective Sensing Framework

This chapter presents the development of original techniques that propose to frame, and to solve, the single point of view data driven selective sensing problem, within a framework adopting the counterbalancing criteria of accuracy and spatial coverage. This occurs through the development of two proposed improvement metrics that are directed independently towards guiding improved accuracy, and improved spatial coverage, with three original selection processes that utilize these improvement metrics. In terms of the accuracy of where features lie in the scene, the goal is to maximize the amount of improvement that can be achieved with the next point to be acquired. The proffered methodology in this thesis to address accuracy is developed in the context of surface meshes using a combination of Delaunay triangulation and Kriging, from which the development of an original expression of improvement is proposed. In terms of improving the spatial coverage of the scene within the sensor's point of view, probabilistic occupancy grids are employed in the development of an original expression for the improvement that a potential future acquisition will make to the overall scene. Furthermore both the proposed surface mesh based improvement and the proposed probabilistic occupancy grid based improvement methods employ reasonable approaches and approximations to ensure that they are computationally efficient.

Figure 3.1 illustrates the framework of the proposed data driven selective sensing algorithm developed in this chapter. Firstly, an initial coarse sampling is performed within the field of view of the sensor. This initial coarse sampling (typically uniformly distributed) is not required for the evaluation of the proposed improvement metrics, but will reduce the number of overall calculations made, while providing an initial stochastic basis to drive future acquisitions. After the acquisition of range measurements, these are integrated concurrently into the surface mesh and the probabilistic occupancy grid models. Included in these models, is the construction of the respective improvement maps, and in order to prevent self-occlusions within the dataset, as discussed in Section 2.4.1, the native coordinate system of the sensor is used for representing the data within each model. The resulting improvement maps are then provided to a point selection process, where the next point is chosen based on providing the most overall scene improvement, which is accuracy, coverage, or a combination of both depending on the specific point selection process in use. This process continues in an iterative and incremental fashion until the desired level of improvement is

achieved, while at the same time resulting in the generation of an estimated depth map, as well as the surface mesh of the scene and the probabilistic occupancy grid representation.

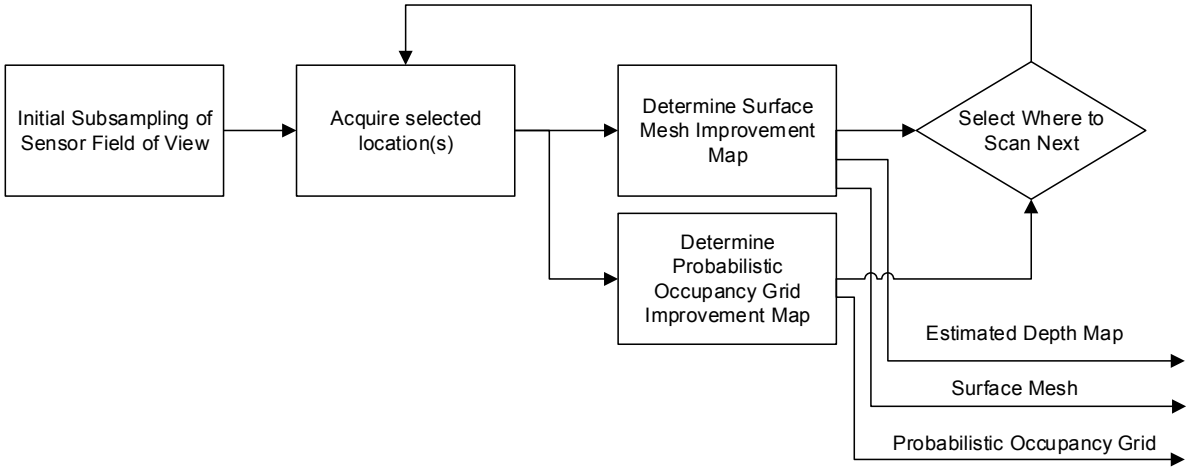


Figure 3.1 - High-level algorithm of intelligent selective sensing as proposed in this thesis

The first section of this chapter covers the development of the proposed surface mesh improvement measure, and the second section covers the development of the proposed probabilistic occupancy grid based improvement measure. The third section covers the development of three original point selection processes. The first is based solely on the surface mesh improvement map that focuses primarily on improving the accuracy of features locations within the scene, the second is based solely on the probabilistic occupancy grid improvement map that drives future acquisitions to improve the spatial coverage of the scene within the sensor's point of view, and the third combines both improvement maps which balances the goal of accurate representation of the scene with ensuring that spatial coverage is also considered. The final section summarizes the theoretical development.

3.1 Surface Mesh Improvement Measure

A surface mesh provides a coarse representation of a 3D scene, and effectively interpolates a scene using planes bounded by the points of the surface geometry captured by the sensor. As strictly plane fitting does not adequately represent complex scenes in many cases, another technique is needed to interpolate within the sensor space. The interpolation should not overly smooth the data, but take into account local statistics. Therefore a global approach is

not desired, and the interpolation approach should also provide an estimate of the error involved in the interpolation. To prevent self-occlusion within the dataset, the native sensor coordinate system will be used to represent the data, as discussed in Section 2.4.1. To ensure that locality of the estimates is maintained, a Delaunay triangulation will be performed on the acquired points. A Delaunay triangulation is used since it ensures that the triangle will not be long and skinny: i.e. triangles will cover a more local region than what other techniques provide. Delaunay triangulation solves the issue of grouping known points together to form structures of locally organized points. With these locally organized points, there needs to be interpolation done between them, which will be accomplished using Ordinary Kriging that was introduced in Section 2.2.6.

The Kriging technique requires many known points in order to more accurately determine the trends of the data due to the law of large numbers. But in order to improve computation time, here the number of known points used in for the Kriging is limited to the vertices of the encompassing triangle, as well as the vertices of the triangles opposite to the encompassing triangle's edges. This reduces the number of known points down to 5 or 6 points of a high degree of locality, as shown later in Figure 3.4. In [43,44,49] it is shown that as a known point used in the Kriging system moves further away from the point of estimation, the weight of that point in the system becomes low. In order to optimize computations, it is typical to drop points that are far away, as their contribution is less than nearby points. Greinstead *et al.* [48] and Haas [49] use this as a justification to perform windowed Kriging, where the area being estimated is broken up into segments where a local window of known points are used to perform the estimations. In this thesis, the number of points used in the Kriging process is minimal, and can be considered as an adaptive window based upon the Delaunay triangulation. The advantage of performing this, is that it increases the contribution of the local values, with the side benefit of reducing the number of computations needed, but at the same time it can lead to a larger estimation error due to the effects that a single measurement error can make on small datasets and the possibility of sharp edges between triangles in the resulting estimation.

Through the use of the described method, an original measure of improvement is developed, inspired from the Ordinary Kriging equations, which will efficiently estimate the potential

improvement in the accuracy of represented features contained in the scene that a potential future acquisition made at currently unmeasured locations within the sensor's field of view will provide. The first subsection describes the semivariogram as used within this technique, followed by the development of the original improvement measure based upon Ordinary Kriging. The improvement measure is developed using Cartesian coordinates, but the application of this improvement measure in the selective sensing algorithm relies on the sensor's native coordinates, with the Cartesian pair $(x,\ y)$ being replaced by the appropriate coordinate pair $(c_0,\ c_1)$, and the Cartesian z being replaced by the native distance coordinate c_2, which is discussed in Section 2.4.1, and presented in more details in Appendix A.

3.1.1 Semivariogram

Kriging requires the analysis of statistical parameters to determine a trend which is used in the interpolation. The method typically used is finding a semivariogram model that fits the empirical semivariance reasonably well, as discussed in Section 2.2.7. Since only a very small number of points are being used, and to ensure the smoothest possible fitting of the resultant model, the semivariogram model used in this thesis will be symmetric, which means that it is only dependant on distance, not direction. By not considering the direction, the variance of the estimated depth will be larger in the case of a defined directional trend in the semivariance, but since Ordinary Kriging already assumes that any trends are negligible in the local neighbourhoods which are being estimated, no additional assumptions are being made, with the net benefit being a much simpler and direct calculation.

The computationally efficient Nugget effect model (described in Section 2.2.7) does not correctly represent the case where uncertainty grows as the point of estimation moves further away from known data points, and as such is a poor choice to use for this application by itself. Furthermore, as all the other semivariogram models listed in Table 2.1 have computational issues in regards to the improvement formulation that the next section presents, mainly that any power of h^w, where $w \neq 2$, is computationally expensive to calculate over many points, they are not ideal choices to directly select for the semivariogram model. The power model, h^w, where $w = 2$, is computationally efficient, but it unfortunately produces an impermissible semivariogram [44,51], which can lead to ill-conditioned and singular matrices. To avoid this limitation, approximation theory is applied

such that the usage of h^w, when $w = 2$, is used as an approximation for the case when $w = 2 - \varepsilon$, for the purposes of developing the improvement measure. This provides the advantage of the mathematical simplicity of the impermissible power model when $w = 2$ for developing the mathematics of the improvement measure, while at the same time using the permissible power model, when $w = 2 - \varepsilon$, and $\varepsilon > 0$, for use in the Kriging calculations. To gain even more computational efficiency, a look up table of h^w between $w = 2$, and $w = 2 - \varepsilon$ will be used, thereby limiting the use of the computationally expensive power operator.

The selection of the value for ε is limited by several factors. The lower bound of ε is limited by the numerical precision of the floating point representation employed while making the calculations, as the produced system may effectively become singular if ε is made too small. The upper bound of ε is limited by the acceptable error that is introduced into the system by the approximations. Figure 3.2 shows the error, $|h^2 - h^{2-\varepsilon}|$, for three different values of ε: 0.1, 0.01, and 0.001. It can be observed that there is an error 'bump' between $0 \leq h \leq 1$, and then the error monotonically increases from zero when $h \geq 1$. For the same values of ε, Figure 3.3 shows the percentage error, $100\frac{|h^2 - h^{2-\varepsilon}|}{h^2}$, which illustrates a different trend, namely that the percentage error grows at a much larger rate as the distance, h, approaches 0 from $h = 1$ than it does when h increases starting from $h = 1$. Note that the minimum percentage error, 0%, occurs when $h = 1$. Figure 3.3 details that as ε decreases from 10^{-1} to 10^{-3}, that the region of where exponential growth of the percentage error occurs for $1 \geq h \geq 0$ shifts to be closer to $h = 0$. This implies that ε should be selected based on the minimum expected distance, such that the region of exponential growth is avoided, in order to maintain a minimal error. Furthermore, in order to reduce error, the value of ε should also be selected based on the maximum absolute error that is to be tolerated into the design of the system, although the growth of the error for larger h is of less importance, since in general, the error in estimation increasing with distance to known values is expected anyways.

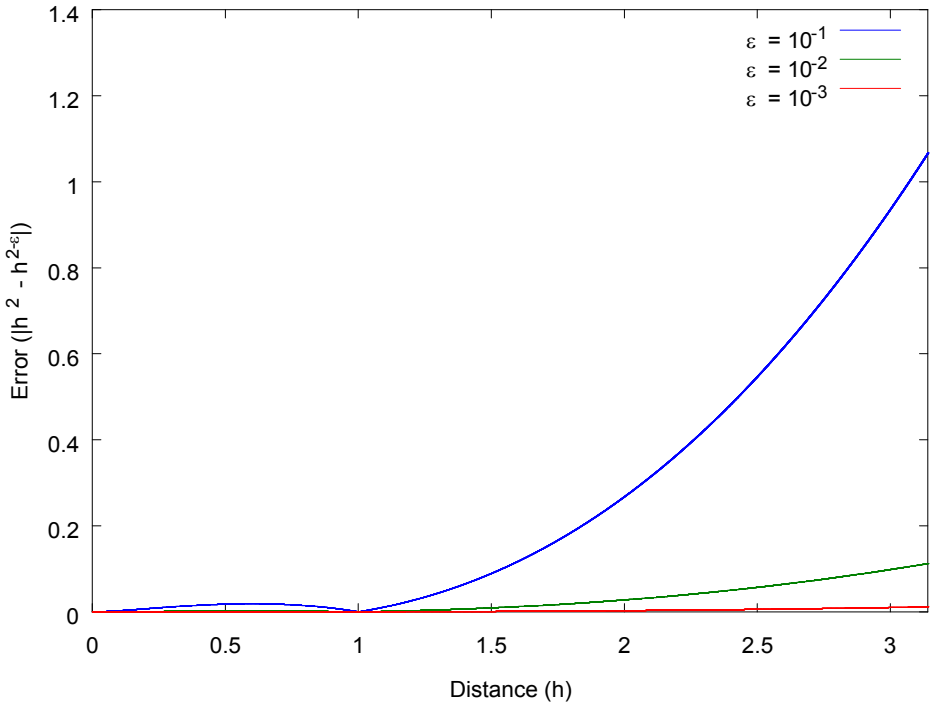


Figure 3.2 - Error between h^2 and $h^{2-\varepsilon}$ for different values of ε

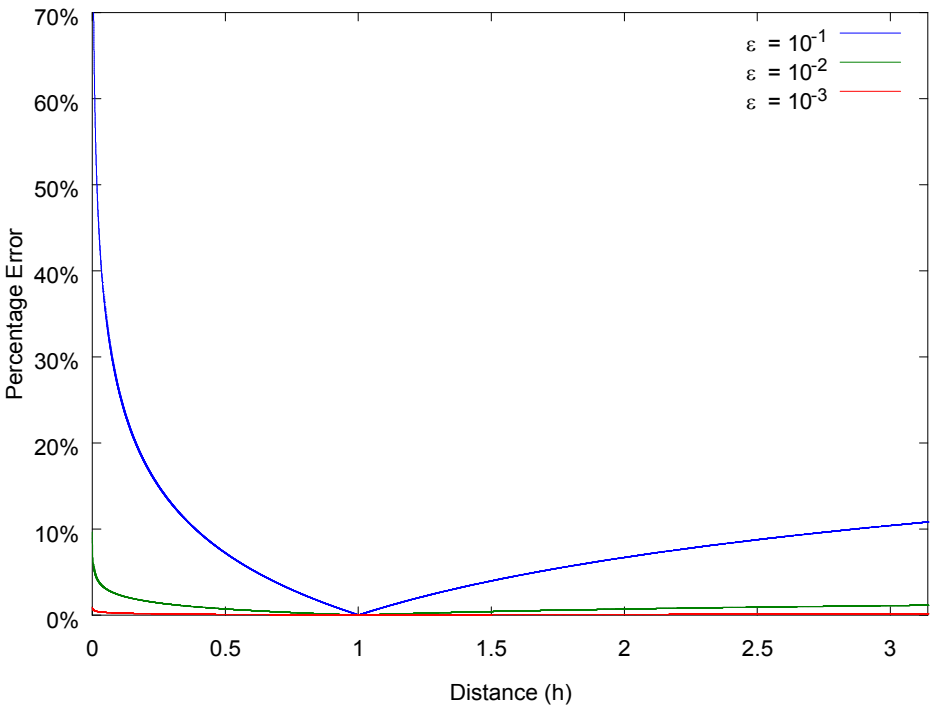


Figure 3.3 - Percentage error between h^2 and $h^{2-\varepsilon}$ for different values of ε

Using the power model by itself for the semivariogram will only capture the trend for variation of depths to increase with distance from the current point, but will not correctly capture a constant variation of depth with respect to distance. Therefore a linear combination between the power model and the nugget model will be performed, which will produce a permissible semivariogram model due to the principle discussed in Section 2.2.7, that any linear combination of permissible semivariogram models will produce a permissible semivariogram model [44]. Therefore the semivariogram model which is being proposed within this context is presented eq. (3.1) with $U(h)$ as defined in eq. (3.2).

$$\gamma(h) = ah^w + bU(h) \tag{3.1}$$

$$U(h) = \begin{cases} 0, & h = 0 \\ 1, & h > 0 \end{cases} \tag{3.2}$$

With the semivariogram model chosen, the empirical semivariogram must be fit for each local region that the Kriging is being performed on. To ensure that the points used are local to the region being Kriged, the choice made is to use only the 3 points of the triangle that encloses the local region, as well as the points opposite to the edges of the local triangle, which leads to five or six points over which the empirical semivariogram is being calculated (see Figure 3.4 for examples of the configuration of points).

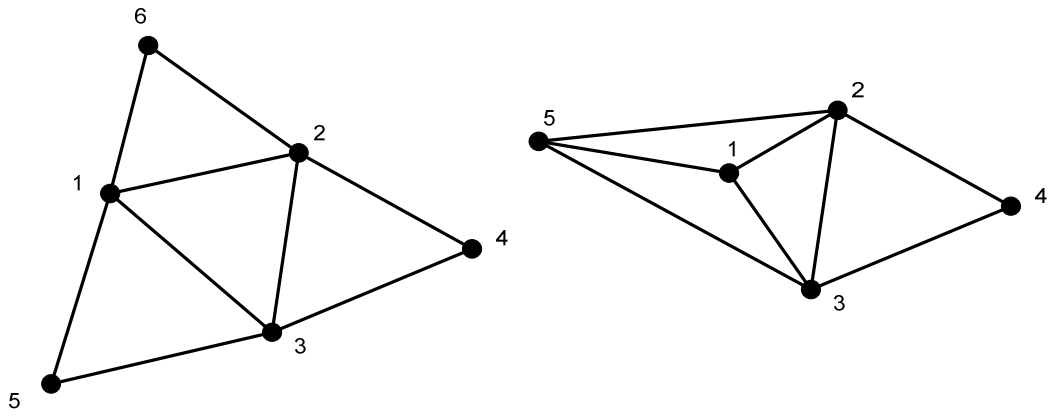


Figure 3.4 - Points selected for Kriging within triangle 1-2-3: general case of 6 points (left) and special case of 5 points (right)

$$h(p_i, p_j) = \sqrt{\left(c_{0_i} - c_{0_j}\right)^2 + \left(c_{1_i} - c_{1_j}\right)^2} \tag{3.3}$$

The empirical semivariogram, $\gamma_{emp}\left(h(p_i,p_j)\right)$, eq. (3.4), is calculated by determining the distance, $h(p_i,p_j)$ as defined by eq. (3.3), and the corresponding semivariance, $\frac{1}{2}\left(z(p_i)-z(p_j)\right)^2$, between each possible pair of points p_i and p_j where $i \neq j$. In the case for n=6 points, this yields 6 choose 2, or more mathematically succinct $\binom{6}{2}=\frac{6!}{2!4!}=15$ possible distance/semivariance pairs, and for n=5 points yields $\binom{5}{2}=\frac{5!}{2!3!}=10$ possible distance/semivariance pairs. From these corresponding distance/semivariance pairs, the parameters a and b of the semivariogram model from eq. (3.1) are determined using traditional least squares fitting (see eq. (3.5)) such that the permissible model can be used in the analysis, where sn is the sum of the number of corresponding distance/semivariance pairs, sh^{2w} is the summation of the squares of h^w, sh^w is the sum of h^w, $s\gamma$ is the sum of the empirical semivariances γ , and $s\gamma h^w$ is the sum of the product of γ and h^w.

$$\gamma_{emp}\left(h(p_i,p_j)\right)=\frac{1}{2}\left(z(p_i)-z(p_j)\right)^2 \tag{3.4}$$

$$\begin{bmatrix}b\\a\end{bmatrix}=\frac{1}{(sn)(sh^{2w})-(sh^w)^2}\begin{bmatrix}sh^{2w} & -sh^w\\ -sh^w & sn\end{bmatrix}\begin{bmatrix}s\gamma\\ s\gamma h^w\end{bmatrix}$$

$$sn=\sum_{i=1}^{n}\sum_{j=i+1}^{n}1$$

$$sh^{2w}=\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(h^w(p_i,p_j)\right)^2$$

$$sh^{w}=\sum_{i=1}^{n}\sum_{j=i+1}^{n}h^w(p_i,p_j) \tag{3.5}$$

$$s\gamma=\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(\frac{1}{2}\left(z(p_i)-z(p_j)\right)^2\right)$$

$$s\gamma h^w=\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(\frac{1}{2}\left(z(p_i)-z(p_j)\right)^2 h^w(p_i,p_j)\right)$$

3.1.2 Surface Mesh Improvement Calculation

The actual improvement resulting from any given acquisition cannot be known until after a new point is acquired, therefore an estimate of this improvement must be determined, since the goal of selective sensing is not to acquire points that would not lead to a significant improvement. The variance to mean ratio (VMR), as defined in eq. (3.6), is selected as the measure of the current error in the estimation. This selection is inspired from the fact that Ordinary Kriging provides both the estimated depth, $\hat{z}(\hat{p}_j)$, from eq. (2.15), as well as the estimated variance on the estimation, $\hat{\sigma}^2(\hat{p}_j)$, from eq. (2.16), for an unmeasured point $\hat{p}_j$. The VMR also appropriately reflects the fact that typically as a depth measurement is collected further from the sensor, the error on the measurement increases, and is inherently normalized in the formulation of the VMR.

$$vmr(\hat{p}_j) = \frac{\hat{\sigma}^2(\hat{p}_j)}{\hat{z}(\hat{p}_j)} \tag{3.6}$$

To calculate an estimate of the measure of improvement that acquiring a particular point will import on the estimated model of the scene, a future estimate of the error must be known. This is accomplished by stating what would the new estimated errors be if the depth measurement, $z(p_s)$, is acquired at some point, p_s, which has not yet been acquired. Since p_s comes from the set of currently unknown points on which the Kriging estimates are currently being performed, let $p_s = \hat{p}_s$ and $z(\hat{p}_s) = \hat{z}(\hat{p}_s)$, then the estimated error measure for $\hat{p}_j$ can be calculated given the hypothesis that the values for $\hat{p}_s$ are known, as shown in eq. (3.7).

$$vmr(\hat{p}_j|\hat{p}_s) = \frac{\hat{\sigma}^2(\hat{p}_j|\hat{p}_s)}{\hat{z}(\hat{p}_j|\hat{p}_s)} \tag{3.7}$$

Now that the estimations of the current and future errors of $\hat{p}_j$ have been calculated given the assumed values for $\hat{p}_s$, the estimated surface mesh (SM) improvement, $imp_{SM}(\hat{p}_j|\hat{p}_s)$, using the error measure of the future value at $\hat{p}_j$ given that the next point acquired is at $\hat{p}_s$ yields the following equation:

$$imp_{SM}(\hat{p}_j|\hat{p}_s) = vmr(\hat{p}_j) - vmr(\hat{p}_j|\hat{p}_s) \tag{3.8}$$

It is not enough to calculate what the improvement of only one point will be if the acquisition of $\hat{p}_s$ is made, but the total improvement for all the m points which $\hat{p}_s$ will affect needs to be calculated. This leads to performing the summation for all points in the local region in order to produce the overall estimated improvement in the results if $\hat{p}_s$ is acquired.

$$imp_{SM}(\hat{p}_s) = \sum_{j=1}^{m} imp_{SM}(\hat{p}_j|\hat{p}_s) \tag{3.9}$$

Using this method, a map of estimated improvements by virtually acquiring at all point $\hat{p}_s$'s can be created, and then this estimated improvement map can be used as a basis for the selection process to decide which point to actually acquire next.

Now that the measure of estimated improvement has been conceptually defined, the mathematical derivation of it is performed. First, to determine the values needed to be solved, the improvement equation is expanded.

$$imp_{SM}(\hat{p}_s) = \sum_{j=1}^{m} \left(vmr(\hat{p}_j) - vmr(\hat{p}_j|\hat{p}_s) \right) \tag{3.10}$$

$$imp_{SM}(\hat{p}_s) = \sum_{j=1}^{m} \left(\frac{\hat{\sigma}^2(\hat{p}_j)}{\hat{z}(\hat{p}_j)} - \frac{\hat{\sigma}^2(\hat{p}_j|\hat{p}_s)}{\hat{z}(\hat{p}_j|\hat{p}_s)} \right) \tag{3.11}$$

The values for $\hat{z}(\hat{p}_j)$ and $\hat{\sigma}^2(\hat{p}_j)$ are obtained from Kriging, so all that is needed is to determine the values for $\hat{z}(\hat{p}_j|\hat{p}_s)$ and $\hat{\sigma}^2(\hat{p}_j|\hat{p}_s)$. This can be determined through inspection of the Ordinary Kriging equations shown previously in the Kriging mathematics, defined in Section 2.2.6.

$$K = \begin{bmatrix} \gamma(h(p_1,p_1)) & \gamma(h(p_1,p_2)) & \cdots & \gamma(h(p_1,p_n)) & 1 \\ \gamma(h(p_2,p_1)) & \gamma(h(p_2,p_2)) & \cdots & \gamma(h(p_2,p_n)) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(h(p_n,p_1)) & \gamma(h(p_n,p_2)) & \cdots & \gamma(h(p_n,p_n)) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \tag{3.12}$$

$$k(\hat{p}_j) = \begin{bmatrix} \gamma\left(h(p_1,\hat{p}_j)\right) \\ \gamma\left(h(p_2,\hat{p}_j)\right) \\ \vdots \\ \gamma\left(h(p_n,\hat{p}_j)\right) \\ 1 \end{bmatrix} \tag{3.13}$$

$$Z = \begin{bmatrix} z(p_1) \\ z(p_2) \\ \vdots \\ z(p_n) \\ 0 \end{bmatrix} \tag{3.14}$$

$$\lambda(\hat{p}_j) = K^{-1}k(\hat{p}_j) \tag{3.15}$$

$$\hat{z}(\hat{p}_j) = \lambda^T(\hat{p}_j)Z \tag{3.16}$$

$$\hat{\sigma}^2(\hat{p}_j) = \lambda^T(\hat{p}_j)k(\hat{p}_j) \tag{3.17}$$

The usage of an isometric semivariogram, γ, means that all that matters is the magnitude of the variation with respect to distance between points, and not the direction of the variation, hence $\gamma\left(h(p_\mathrm{i}, p_\mathrm{j})\right) = \gamma\left(h(p_\mathrm{j}, p_\mathrm{i})\right)$. This, in turn, leads to the fact that K will be a symmetric matrix, $K = K^T$. The K matrix for the case in which it is assumed that $\hat{p}_s$ is known, which will be defined as $\widehat{K}(\hat{p}_s)$, and will be an $(n+2) \times (n+2)$ matrix, where n is the number of known points, is as follows:

$$\widehat{K}(\hat{p}_s) = \begin{bmatrix} \gamma(h(p_1,p_1)) & \gamma(h(p_1,p_2)) & \cdots & \gamma(h(p_1,p_n)) & \gamma(h(p_1,\hat{p}_s)) & 1 \\ \gamma(h(p_2,p_1)) & \gamma(h(p_2,p_2)) & \cdots & \gamma(h(p_2,p_n)) & \gamma(h(p_2,\hat{p}_s)) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma(h(p_n,p_1)) & \gamma(h(p_n,p_2)) & \cdots & \gamma(h(p_n,p_n)) & \gamma(h(p_n,\hat{p}_s)) & 1 \\ \gamma(h(\hat{p}_s,p_1)) & \gamma(h(\hat{p}_s,p_2)) & \cdots & \gamma(h(\hat{p}_s,p_n)) & \gamma(h(\hat{p}_s,\hat{p}_s)) & 1 \\ 1 & 1 & \cdots & 1 & 1 & 0 \end{bmatrix} \tag{3.18}$$

And similarly the rest of the Kriging system corresponding to this situation will be:

$$\hat{k}(\hat{p}_j|\hat{p}_s) = \begin{bmatrix} \gamma\left(h(p_1,\hat{p}_j)\right) \\ \gamma\left(h(p_2,\hat{p}_j)\right) \\ \vdots \\ \gamma\left(h(p_n,\hat{p}_j)\right) \\ \gamma\left(h(\hat{p}_s,\hat{p}_j)\right) \\ 1 \end{bmatrix} \tag{3.19}$$

$$\hat{Z}(\hat{p}_s) = \begin{bmatrix} z(p_1) \\ z(p_2) \\ \vdots \\ z(p_n) \\ z(\hat{p}_s) \\ 0 \end{bmatrix} \tag{3.20}$$

$$\lambda(\hat{p}_j|\hat{p}_s) = \widehat{K}^{-1}(\hat{p}_s)\hat{k}(\hat{p}_j|\hat{p}_s) \tag{3.21}$$

$$\hat{z}(\hat{p}_j|\hat{p}_s) = \lambda^T(\hat{p}_j|\hat{p}_s)\hat{Z}(\hat{p}_s) \tag{3.22}$$

$$\hat{\sigma}^2(\hat{p}_j|\hat{p}_s) = \lambda^T(\hat{p}_j|\hat{p}_s)\hat{k}(\hat{p}_j|\hat{p}_s) \tag{3.23}$$

It would be computationally expensive to have to calculate these values for each point in the unknown data space, so to make these computationally tractable, and allow maximal reuse of data, these terms will be expressed with respect to the original Kriging terms. Through the inspection of eq. (3.18), it is obvious that the $(n+1)$ column of $\widehat{K}(\hat{p}_s)$ contains the same terms as $k(\hat{p}_s)$ would with the addition of one more entry, namely that of $\gamma\big(h(\hat{p}_s, \hat{p}_s)\big)$, and since $\widehat{K}(\hat{p}_s)$ is a symmetric matrix due to the isometric semivariogram the $(n+1)$ row of $\widehat{K}(\hat{p}_s)$ can be represented by $k^T(\hat{p}_s)$ with the addition of the $\gamma\big(h(\hat{p}_s, \hat{p}_s)\big)$ term. Similarly $\hat{k}\big(\hat{p}_j|\hat{p}_s\big)$ contains the same terms as $k\big(\hat{p}_j\big)$ with the additional entry of $\gamma\left(h\big(\hat{p}_s, \hat{p}_j\big)\right)$. Using this knowledge, $\widehat{K}'(\hat{p}_s)$, $\hat{k}'\big(\hat{p}_j|\hat{p}_s\big)$, $\hat{Z}'(\hat{p}_s)$, and $\lambda'\big(\hat{p}_j|\hat{p}_s\big)$ can be defined by using a permutation matrix to swap the second last row and column with the last row and column in $\widehat{K}(\hat{p}_s)$, $\hat{k}\big(\hat{p}_j|\hat{p}_s\big)$, $\hat{Z}(\hat{p}_s)$, and $\lambda\big(\hat{p}_j|\hat{p}_s\big)$ respectively such that the new matrices can be represented in a partition form of previously calculated values. This $(n+2)\times(n+2)$ permutation matrix, as shown in eq. (3.24), where $I_{n\times n}$ is the $n \times n$ identity matrix, and $\emptyset_{i\times j}$ is the $i \times j$ null matrix.

$$P = \begin{bmatrix} I_{n\times n} & \emptyset_{n\times 2} \\ \emptyset_{2\times n} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{bmatrix} \tag{3.24}$$

Therefore, the permutated Kriging system becomes:

$$\begin{aligned} \widehat{K}'(\hat{p}_s) &= P\widehat{K}(\hat{p}_s)P^T \\ &= \begin{bmatrix} \gamma\big(h(p_1,p_1)\big) & \gamma\big(h(p_1,p_2)\big) & \cdots & \gamma\big(h(p_1,p_n)\big) & 1 & \gamma\big(h(p_1,\hat{p}_s)\big) \\ \gamma\big(h(p_2,p_1)\big) & \gamma\big(h(p_2,p_2)\big) & \cdots & \gamma\big(h(p_2,p_n)\big) & 1 & \gamma\big(h(p_2,\hat{p}_s)\big) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma\big(h(p_n,p_1)\big) & \gamma\big(h(p_n,p_2)\big) & \cdots & \gamma\big(h(p_n,p_n)\big) & 1 & \gamma\big(h(p_n,\hat{p}_s)\big) \\ 1 & 1 & \cdots & 1 & 0 & 1 \\ \gamma\big(h(\hat{p}_s,p_1)\big) & \gamma\big(h(\hat{p}_s,p_2)\big) & \cdots & \gamma\big(h(\hat{p}_s,p_n)\big) & 1 & \gamma\big(h(\hat{p}_s,\hat{p}_s)\big) \end{bmatrix} \\ &= \begin{bmatrix} K & k(\hat{p}_s) \\ k^T(\hat{p}_s) & \gamma\big(h(\hat{p}_s,\hat{p}_s)\big) \end{bmatrix} \end{aligned} \tag{3.25}$$

$$\hat{k}'(\hat{p}_j|\hat{p}_s) = P\hat{k}(\hat{p}_j|\hat{p}_s) = \begin{bmatrix} \gamma\left(h(p_1,\hat{p}_j)\right) \\ \gamma\left(h(p_2,\hat{p}_j)\right) \\ \vdots \\ \gamma\left(h(p_n,\hat{p}_j)\right) \\ 1 \\ \gamma\left(h(\hat{p}_s,\hat{p}_j)\right) \end{bmatrix} = \begin{bmatrix} k(\hat{p}_j) \\ \gamma\left(h(\hat{p}_s,\hat{p}_j)\right) \end{bmatrix} \tag{3.26}$$

$$\hat{Z}'(\hat{p}_s) = P\hat{Z}(\hat{p}_s) = \begin{bmatrix} z(p_1) \\ z(p_2) \\ \vdots \\ z(p_n) \\ 0 \\ z(\hat{p}_s) \end{bmatrix} = \begin{bmatrix} Z \\ z(\hat{p}_s) \end{bmatrix} \tag{3.27}$$

Using these derived matrices, $\hat{z}(\hat{p}_j|\hat{p}_s)$ and $\hat{\sigma}^2(\hat{p}_j|\hat{p}_s)$ can be determined, and it can be additionally shown that $\hat{z}'(\hat{p}_j|\hat{p}_s) = \hat{z}(\hat{p}_j|\hat{p}_s)$, and $\hat{\sigma}'^2(\hat{p}_j|\hat{p}_s) = \hat{\sigma}^2(\hat{p}_j|\hat{p}_s)$, which means that the order of the known values does not matter for Kriging.

$$\lambda'(\hat{p}_j|\hat{p}_s) = \hat{K}'^{-1}(\hat{p}_s)\hat{k}'(\hat{p}_j|\hat{p}_s) \tag{3.28}$$

$$\lambda'(\hat{p}_j|\hat{p}_s) = P^{T^{-1}}\hat{K}^{-1}(\hat{p}_s)P^{-1}\,P\hat{k}(\hat{p}_j|\hat{p}_s) = P^{T^{-1}}\hat{K}^{-1}(\hat{p}_s)\hat{k}(\hat{p}_j|\hat{p}_s) = P^{T^{-1}}\lambda(\hat{p}_j|\hat{p}_s) \tag{3.29}$$

$$\hat{z}'(\hat{p}_j|\hat{p}_s) = \lambda'^T(\hat{p}_j|\hat{p}_s)\hat{Z}'(\hat{p}_s) = \lambda^T(\hat{p}_j|\hat{p}_s)P^{-1}P\hat{Z}(\hat{p}_s) = \lambda^T(\hat{p}_j|\hat{p}_s)\hat{Z}(\hat{p}_s) = \hat{z}(\hat{p}_j|\hat{p}_s) \tag{3.30}$$

$$\begin{aligned}\hat{\sigma}'^2(\hat{p}_j|\hat{p}_s) &= \lambda'^T(\hat{p}_j|\hat{p}_s)\hat{k}'(\hat{p}_j|\hat{p}_s) = \lambda^T(\hat{p}_j|\hat{p}_s)P^{-1}P\hat{k}(\hat{p}_j|\hat{p}_s) = \lambda^T(\hat{p}_j|\hat{p}_s)\hat{k}(\hat{p}_j|\hat{p}_s) \\ &= \hat{\sigma}^2(\hat{p}_j|\hat{p}_s)\end{aligned} \tag{3.31}$$

With the equations above, it has been confirmed that the permutation matrix does not change the results of the Kriging system, and hence solving for $\hat{\sigma}^2(\hat{p}_j|\hat{p}_s)$ and $\hat{z}(\hat{p}_j|\hat{p}_s)$ can be performed using the permutated and partitioned Kriging system.

$$\hat{\sigma}^2(\hat{p}_j|\hat{p}_s) = \lambda'^T(\hat{p}_j|\hat{p}_s)\hat{k}'(\hat{p}_j|\hat{p}_s) \tag{3.32}$$

By substituting in the formula for $\lambda'^T(\hat{p}_j|\hat{p}_s)$ (eq. (3.28)), and taking into account that $\hat{K}^T(\hat{p}_s) = \hat{K}(\hat{p}_s)$, and hence $\hat{K}^{-T}(\hat{p}_s) = \hat{K}^{-1}(\hat{p}_s)$, into eq. (3.32), the following equation is obtained:

$$\begin{aligned}\hat{\sigma}^2(\hat{p}_j|\hat{p}_s) &= \left(\hat{K}'^{-1}(\hat{p}_s)\hat{k}'(\hat{p}_j|\hat{p}_s)\right)^{\mathrm{T}}\hat{k}'(\hat{p}_j|\hat{p}_s) = \hat{k}'^T(\hat{p}_j|\hat{p}_s)\hat{K}'^{-\mathrm{T}}(\hat{p}_s)\hat{k}'(\hat{p}_j|\hat{p}_s) \\ &= \hat{k}'^T(\hat{p}_j|\hat{p}_s)\hat{K}'^{-1}(\hat{p}_s)\hat{k}'(\hat{p}_j|\hat{p}_s)\end{aligned} \tag{3.33}$$

Substituting in the block matrix forms of the expressions:

$$\hat{\sigma}^2(\hat{p}_j|\hat{p}_s) = \begin{bmatrix} k(\hat{p}_j) \\ \gamma\left(h(\hat{p}_s,\hat{p}_j)\right) \end{bmatrix}^T \begin{bmatrix} K & k(\hat{p}_s) \\ k^T(\hat{p}_s) & \gamma(h(\hat{p}_s,\hat{p}_s)) \end{bmatrix}^{-1} \begin{bmatrix} k(\hat{p}_j) \\ \gamma\left(h(\hat{p}_s,\hat{p}_j)\right) \end{bmatrix} \tag{3.34}$$

The matrix, $\begin{bmatrix} K & k(\hat{p}_s) \\ k^T(\hat{p}_s) & \gamma(h(\hat{p}_s,\hat{p}_s)) \end{bmatrix}^{-1}$, can be inverted using block matrix inversion as defined by eq. (B.1) in Appendix B.

$$\begin{bmatrix} K & k(\hat{p}_s) \\ k^T(\hat{p}_s) & \gamma(h(\hat{p}_s,\hat{p}_s)) \end{bmatrix}^{-1} = \begin{bmatrix} K^{-1} + K^{-1}k(\hat{p}_s)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - k^T(\hat{p}_s)K^{-1}k(\hat{p}_s)\right)^{-1}k^T(\hat{p}_s)K^{-1} & -K^{-1}k(\hat{p}_s)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - k^T(\hat{p}_s)K^{-1}k(\hat{p}_s)\right)^{-1} \\ -\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - k^T(\hat{p}_s)K^{-1}k(\hat{p}_s)\right)^{-1}k^T(\hat{p}_s)K^{-1} & \left(\gamma(h(\hat{p}_s,\hat{p}_s)) - k^T(\hat{p}_s)K^{-1}k(\hat{p}_s)\right)^{-1} \end{bmatrix} \tag{3.35}$$

Using the Kriging formula's for $\hat{\sigma}^2(\hat{p}_s)$ from eq. (3.17) and $\lambda(\hat{p}_s)$ from eq. (3.15), the above expression can be simplified to be:

$$\begin{bmatrix} K & k(\hat{p}_s) \\ k^T(\hat{p}_s) & \gamma(h(\hat{p}_s,\hat{p}_s)) \end{bmatrix}^{-1} = \begin{bmatrix} K^{-1} + \lambda(\hat{p}_s)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s) & -\lambda(\hat{p}_s)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} \\ -\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s) & \left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} \end{bmatrix} \tag{3.36}$$

Therefore:

$$\hat{\sigma}^2(\hat{p}_j|\hat{p}_s) = \begin{bmatrix} k(\hat{p}_j) \\ \gamma\left(h(\hat{p}_s,\hat{p}_j)\right) \end{bmatrix}^T \begin{bmatrix} K^{-1} + \lambda(\hat{p}_s)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s) & -\lambda(\hat{p}_s)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} \\ -\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s) & \left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} \end{bmatrix} \begin{bmatrix} k(\hat{p}_j) \\ \gamma\left(h(\hat{p}_s,\hat{p}_j)\right) \end{bmatrix} \tag{3.37}$$

$$\begin{aligned} \hat{\sigma}^2(\hat{p}_j|\hat{p}_s) = {} & k^T(\hat{p}_j)K^{-1}k(\hat{p}_j) + k^T(\hat{p}_j)\lambda(\hat{p}_s)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s)\,k(\hat{p}_j) \\ & - k^T(\hat{p}_j)\lambda(\hat{p}_s)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\gamma\left(h(\hat{p}_s,\hat{p}_j)\right) \\ & - \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s)k(\hat{p}_j) \\ & + \gamma^2\left(h(\hat{p}_s,\hat{p}_j)\right)\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} \end{aligned} \tag{3.38}$$

Substituting in the equations from the original Kriging system into eq. (3.38), and taking into account that $\left(\gamma(h(\hat{p}_s,\hat{p}_s)) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}$, $k^T(\hat{p}_j)\lambda(\hat{p}_s)$, and $\lambda^T(\hat{p}_s)\,k(\hat{p}_j)$ are scalars, and that $\lambda^T(\hat{p}_s)\,k(\hat{p}_j) = \left(k^T(\hat{p}_j)\lambda(\hat{p}_s)\right)^T$, the following relationship is found:

$$\hat{\sigma}^2(\hat{p}_j|\hat{p}_s) = \hat{\sigma}^2(\hat{p}_j) + \left(k^T(\hat{p}_j)\lambda(\hat{p}_s)\right)^2 \left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} - 2\gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\left(k^T(\hat{p}_j)\lambda(\hat{p}_s)\right)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} + \gamma^2\left(h(\hat{p}_s,\hat{p}_j)\right)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} \tag{3.39}$$

which reduces down to:

$$\hat{\sigma}^2(\hat{p}_j|\hat{p}_s) = \hat{\sigma}^2(\hat{p}_j) + \frac{\left(k^T(\hat{p}_j)\lambda(\hat{p}_s) - \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\right)^2}{\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)} \tag{3.40}$$

Similarly, the calculation for $\hat{z}(\hat{p}_j|\hat{p}_s)$:

$$\hat{z}(\hat{p}_j|\hat{p}_s) = \lambda'^T(\hat{p}_j|\hat{p}_s)\hat{Z}'(\hat{p}_s) \tag{3.41}$$

$$\hat{z}(\hat{p}_j|\hat{p}_s) = \hat{k}'^T(\hat{p}_j|\hat{p}_s)\hat{K}'^{-1}(\hat{p}_s)\hat{Z}'(\hat{p}_s) \tag{3.42}$$

$$\hat{z}(\hat{p}_j|\hat{p}_s) = \begin{bmatrix} k(\hat{p}_j) \\ \gamma\left(h(\hat{p}_s,\hat{p}_j)\right) \end{bmatrix}^T \begin{bmatrix} K^{-1} + \lambda(\hat{p}_s)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s) & -\lambda(\hat{p}_s)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} \\ -\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s) & \left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} \end{bmatrix} \begin{bmatrix} Z \\ z(\hat{p}_s) \end{bmatrix} \tag{3.43}$$

$$\begin{aligned} \hat{z}(\hat{p}_j|\hat{p}_s) = {} & k^T(\hat{p}_j)K^{-1}Z + k^T(\hat{p}_j)\lambda(\hat{p}_s)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s)Z \\ & - k^T(\hat{p}_j)\lambda(\hat{p}_s)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} z(\hat{p}_s) \\ & - \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\lambda^T(\hat{p}_s)Z \\ & + \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} z(\hat{p}_s) \end{aligned} \tag{3.44}$$

After performing substitutions and simplifications:

$$\begin{aligned} \hat{z}(\hat{p}_j|\hat{p}_s) = {} & \hat{z}(\hat{p}_j) + k^T(\hat{p}_j)\lambda(\hat{p}_s)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\hat{z}(\hat{p}_s) \\ & - k^T(\hat{p}_j)\lambda(\hat{p}_s)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} z(\hat{p}_s) \\ & - \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1}\hat{z}(\hat{p}_s) \\ & + \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)^{-1} z(\hat{p}_s) \end{aligned} \tag{3.45}$$

$$\begin{aligned} & \hat{z}(\hat{p}_j|\hat{p}_s) \\ & = \hat{z}(\hat{p}_j) + \frac{k^T(\hat{p}_j)\lambda(\hat{p}_s)\hat{z}(\hat{p}_s) - k^T(\hat{p}_j)\lambda(\hat{p}_s)z(\hat{p}_s) - \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\hat{z}(\hat{p}_s) + \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)z(\hat{p}_s)}{\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)} \end{aligned} \tag{3.46}$$

$$\hat{z}(\hat{p}_j|\hat{p}_s) = \hat{z}(\hat{p}_j) + \frac{\left(k^T(\hat{p}_j)\lambda(\hat{p}_s) - \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\right)\left(\hat{z}(\hat{p}_s) - z(\hat{p}_s)\right)}{\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)} \tag{3.47}$$

Note that the consequence of the eq. (3.47) above, is that if it is assumed $\hat{z}(\hat{p}_s) = z(\hat{p}_s)$, then it follows that $\hat{z}(\hat{p}_j|\hat{p}_s) = \hat{z}(\hat{p}_j)$. If the substitution of the resulting relationships from eq. (3.40) and eq. (3.47) into the improvement equation, eq. (3.11), is made, then the relationship in eq. (3.50) can be developed.

$$imp_{SM}(\hat{p}_s) = \sum_{j=1}^{m} \frac{\hat{\sigma}^2(\hat{p}_j)}{\hat{z}(\hat{p}_j)} - \frac{\hat{\sigma}^2(\hat{p}_j|\hat{p}_s)}{\hat{z}(\hat{p}_j|\hat{p}_s)} \tag{3.48}$$

$$imp_{SM}(\hat{p}_s) = \sum_{j=1}^{m} \frac{\hat{\sigma}^2(\hat{p}_j)}{\hat{z}(\hat{p}_j)} - \frac{\hat{\sigma}^2(\hat{p}_j) + \frac{\left(k^T(\hat{p}_j)\lambda(\hat{p}_s) - \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\right)^2}{\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)}}{\hat{z}(\hat{p}_j)} \tag{3.49}$$

$$imp_{SM}(\hat{p}_s) = -\sum_{j=1}^{m} \frac{\left(k^T(\hat{p}_j)\lambda(\hat{p}_s) - \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\right)^2}{\hat{z}(\hat{p}_j)\left(\gamma\left(h(\hat{p}_s,\hat{p}_s)\right) - \hat{\sigma}^2(\hat{p}_s)\right)} \tag{3.50}$$

Using the fact that $h(\hat{p}_s,\hat{p}_s) = 0$, and $\gamma(0) = 0$, by the definition of the semivariogram into account, the improvement equation can be restated as:

$$imp_{SM}(\hat{p}_s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)} \sum_{j=1}^{m} \frac{\left(k^T(\hat{p}_j)\lambda(\hat{p}_s) - \gamma\left(h(\hat{p}_s,\hat{p}_j)\right)\right)^2}{\hat{z}(\hat{p}_j)} \tag{3.51}$$

Unfortunately in this nice simplified form, the improvement calculation is expensive to calculate. This is due to the fact that the improvement equation needs to be calculated for all m point $\hat{p}_s$'s and the summation being over m terms that are not necessarily equal, means that the inner term is required to be calculated m^2 times. This is where the benefit of the choice in the semivariogram model, $\gamma\left(h(\hat{p}_s,\hat{p}_j)\right) = ah^w(\hat{p}_s,\hat{p}_j) + bU\left(h(\hat{p}_s,\hat{p}_j)\right)$, introduced in eq. (3.1), becomes apparent. If the approximation of the semivariogram model of $w = 2$ is made, as stated in Section 3.1.1, then an expression that is linear with respect of the number of unknown points can be formulated.

Firstly, the expression of the approximate semivariogram is rewritten such that terms with s and j have limited dependence, this is done by substituting in and expanding the distance

function, $h(\hat{p}_s, \hat{p}_j)$, as well as the unit step function, $U(h)$. Since the development of the surface mesh improvement measure is being performed in Cartesian coordinates, where $c_0 = x$ and $c_1 = y$, hence the formulation of eq. (3.52).

$$h(\hat{p}_s, \hat{p}_j) = \sqrt[2]{(\hat{x}_s - \hat{x}_j)^2 + (\hat{y}_s - \hat{y}_j)^2} \tag{3.52}$$

$$U\left(h(\hat{p}_s, \hat{p}_j)\right) = \begin{cases} 0, & h(\hat{p}_s, \hat{p}_j) = 0 \\ 1, & h(\hat{p}_s, \hat{p}_j) > 0 \end{cases} \tag{3.53}$$

$$\begin{aligned} \gamma\left(h(\hat{p}_s, \hat{p}_j)\right) &= ah^2(\hat{p}_s, \hat{p}_j) + bU\left(h(\hat{p}_s, \hat{p}_j)\right) \\ &= \begin{cases} ah^2(\hat{p}_s, \hat{p}_j) + b, & \hat{p}_s \neq \hat{p}_j \\ 0, & \hat{p}_s = \hat{p}_j \end{cases} \\ &= \begin{cases} a\left((\hat{x}_s - \hat{x}_j)^2 + (\hat{y}_s - \hat{y}_j)^2\right) + b, & \hat{p}_s \neq \hat{p}_j \\ 0, & \hat{p}_s = \hat{p}_j \end{cases} \\ &= \begin{cases} a\left(({\hat{x}_s}^2 + {\hat{y}_s}^2) + ({\hat{x}_j}^2 + {\hat{y}_j}^2) - 2\hat{x}_s\hat{x}_j - 2\hat{y}_s\hat{y}_j\right) + b, & \hat{p}_s \neq \hat{p}_j \\ 0, & \hat{p}_s = \hat{p}_j \end{cases} \end{aligned} \tag{3.54}$$

The consequence of substituting this modified semivariogram expression into the improvement equation is as follows:

$$\begin{aligned} imp_{SM}(\hat{p}_s) &= \frac{1}{\hat{\sigma}^2(\hat{p}_s)} \sum_{j=1}^{m} \frac{\left(k^T(\hat{p}_j)\lambda(\hat{p}_s) - \gamma\left(h(\hat{p}_s, \hat{p}_j)\right)\right)^2}{\hat{z}(\hat{p}_j)} \\ &= \frac{1}{\hat{\sigma}^2(\hat{p}_s)} \left(\sum_{j=1, j\neq s}^{m} \frac{\left(k^T(\hat{p}_j)\lambda(\hat{p}_s) - ah^2(\hat{p}_s, \hat{p}_j) - b\right)^2}{\hat{z}(\hat{p}_j)} + \frac{\left(cov(\hat{p}_s, \hat{p}_s)\right)^2}{\hat{z}(\hat{p}_s)} \right) \\ &= \frac{1}{\hat{\sigma}^2(\hat{p}_s)} \left(\sum_{j=1, j\neq s}^{m} \frac{\left(k^T(\hat{p}_j)\lambda(\hat{p}_s)\right)^2 - 2k^T(\hat{p}_j)\lambda(\hat{p}_s)ah^2(\hat{p}_s, \hat{p}_j) - 2k^T(\hat{p}_j)\lambda(\hat{p}_s)b + a^2h^4(\hat{p}_s, \hat{p}_j) + b^2 + 2abh^2(\hat{p}_s, \hat{p}_j)}{\hat{z}(\hat{p}_j)} \right. \\ &\quad \left. + \frac{\hat{\sigma}^4(\hat{p}_s)}{\hat{z}(\hat{p}_s)} \right) \\ &= \frac{1}{\hat{\sigma}^2(\hat{p}_s)} \left(\sum_{j=1, j\neq s}^{m} \frac{\left(k^T(\hat{p}_j)\lambda(\hat{p}_s)\right)^2}{\hat{z}(\hat{p}_j)} - 2a \sum_{j=1, j\neq s}^{m} \frac{k^T(\hat{p}_j)\lambda(\hat{p}_s)h^2(\hat{p}_s, \hat{p}_j)}{\hat{z}(\hat{p}_j)} - 2b \sum_{j=1, j\neq s}^{m} \frac{k^T(\hat{p}_j)\lambda(\hat{p}_s)}{\hat{z}(\hat{p}_j)} + a^2 \sum_{j=1, j\neq s}^{m} \frac{h^4(\hat{p}_s, \hat{p}_j)}{\hat{z}(\hat{p}_j)} \right. \\ &\quad \left. + b^2 \sum_{j=1, j\neq s}^{m} \frac{1}{\hat{z}(\hat{p}_j)} + 2ab \sum_{j=1, j\neq s}^{m} \frac{h^2(\hat{p}_s, \hat{p}_j)}{\hat{z}(\hat{p}_j)} \right) + \frac{\hat{\sigma}^2(\hat{p}_s)}{\hat{z}(\hat{p}_s)} \end{aligned} \tag{3.55}$$

Now by substituting in the known expression for $h^2(\hat{p}_s, \hat{p}_j)$ and by further separating the j from the s terms, the following expressions can be developed:

$$
\begin{aligned}
imp_{SM}(\hat{p}_s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)}\Bigg(& \sum_{j=1, j\neq s}^{m} \frac{\left(k^T(\hat{p}_j)\,\lambda(\hat{p}_s)\right)^T \left(k^T(\hat{p}_j)\,\lambda(\hat{p}_s)\right)}{\hat{z}(\hat{p}_j)} \\
& - 2a \sum_{j=1, j\neq s}^{m} \frac{k^T(\hat{p}_j)\,\lambda(\hat{p}_s)\left(\left(\hat{x}_s{}^2 + \hat{y}_s{}^2\right) + \left(\hat{x}_j{}^2 + \hat{y}_j{}^2\right) - 2\hat{x}_s\hat{x}_j - 2\hat{y}_s\hat{y}_j\right)}{\hat{z}(\hat{p}_j)} - 2b \sum_{j=1, j\neq s}^{m} \frac{k^T(\hat{p}_j)\,\lambda(\hat{p}_s)}{\hat{z}(\hat{p}_j)} \\
& + a^2 \sum_{j=1, j\neq s}^{m} \frac{\left(\left(\hat{x}_s{}^2 + \hat{y}_s{}^2\right) + \left(\hat{x}_j{}^2 + \hat{y}_j{}^2\right) - 2\hat{x}_s\hat{x}_j - 2\hat{y}_s\hat{y}_j\right)^2}{\hat{z}(\hat{p}_j)} + b^2 \sum_{j=1, j\neq s}^{m} \frac{1}{\hat{z}(\hat{p}_j)} \\
& + 2ab \sum_{j=1, j\neq s}^{m} \frac{\left(\left(\hat{x}_s{}^2 + \hat{y}_s{}^2\right) + \left(\hat{x}_j{}^2 + \hat{y}_j{}^2\right) - 2\hat{x}_s\hat{x}_j - 2\hat{y}_s\hat{y}_j\right)}{\hat{z}(\hat{p}_j)}\Bigg) + \frac{\hat{\sigma}^2(\hat{p}_s)}{\hat{z}(\hat{p}_S)}
\end{aligned}
\tag{3.56}
$$

$$
\begin{aligned}
imp_{SM}(\hat{p}_s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)}\Bigg(& \sum_{j=1, j\neq s}^{m} \frac{\lambda^T(\hat{p}_s)k(\hat{p}_j)k^T(\hat{p}_j)\,\lambda(\hat{p}_s)}{\hat{z}(\hat{p}_j)} \\
& - 2a \sum_{j=1, j\neq s}^{m} \frac{k^T(\hat{p}_j)\left(\left(\hat{x}_s{}^2 + \hat{y}_s{}^2\right) + \left(\hat{x}_j{}^2 + \hat{y}_j{}^2\right) - 2\hat{x}_s\hat{x}_j - 2\hat{y}_s\hat{y}_j\right)\lambda(\hat{p}_s)}{\hat{z}(\hat{p}_j)} - 2b \sum_{j=1, j\neq s}^{m} \frac{k^T(\hat{p}_j)\,\lambda(\hat{p}_s)}{\hat{z}(\hat{p}_j)} \\
& + a^2 \sum_{j=1, j\neq s}^{m} \Bigg(\frac{\left(\hat{x}_s{}^2 + \hat{y}_s{}^2\right)^2 + 2\left(\hat{x}_s{}^2 + \hat{y}_s{}^2\right)\left(\hat{x}_j{}^2 + \hat{y}_j{}^2\right) - 4\left(\hat{x}_s{}^2 + \hat{y}_s{}^2\right)\hat{x}_s\hat{x}_j - 4\left(\hat{x}_s{}^2 + \hat{y}_s{}^2\right)\hat{y}_s\hat{y}_j}{\hat{z}(\hat{p}_j)} \\
& + \frac{\left(\hat{x}_j{}^2 + \hat{y}_j{}^2\right)^2 - 4\hat{x}_s\left(\hat{x}_j{}^2 + \hat{y}_j{}^2\right)\hat{x}_j - 4\hat{y}_s\left(\hat{x}_j{}^2 + \hat{y}_j{}^2\right)\hat{y}_j + 4\hat{x}_s{}^2\hat{x}_j{}^2 + 8\hat{x}_s\hat{y}_s\hat{x}_j\hat{y}_j + 4\hat{y}_s{}^2\hat{y}_j{}^2}{\hat{z}(\hat{p}_j)}\Bigg) \\
& + b^2 \sum_{j=1, j\neq s}^{m} \frac{1}{\hat{z}(\hat{p}_j)} + 2ab \sum_{j=1, j\neq s}^{m} \frac{\left(\hat{x}_s{}^2 + \hat{y}_s{}^2\right) + \left(\hat{x}_j{}^2 + \hat{y}_j{}^2\right) - 2\hat{x}_s\hat{x}_j - 2\hat{y}_s\hat{y}_j}{\hat{z}(\hat{p}_j)}\Bigg) + \frac{\hat{\sigma}^2(\hat{p}_s)}{\hat{z}(\hat{p}_S)}
\end{aligned}
\tag{3.57}
$$

$$
\begin{aligned}
imp_{SM}(\hat{p}_s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)}\Bigg(& \lambda^T(\hat{p}_s)\left(\sum_{j=1,j\neq s}^{m}\frac{k(\hat{p}_j)k^T(\hat{p}_j)}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s) \\
& - 2a\Bigg(\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)\left(\sum_{j=1,j\neq s}^{m}\frac{k^T(\hat{p}_j)}{\hat{z}(\hat{p}_j)}\right)+\left(\sum_{j=1,j\neq s}^{m}\frac{k^T(\hat{p}_j)(\hat{x}_j{}^2+\hat{y}_j{}^2)}{\hat{z}(\hat{p}_j)}\right)-2\hat{x}_s\left(\sum_{j=1,j\neq s}^{m}\frac{k^T(\hat{p}_j)\hat{x}_j}{\hat{z}(\hat{p}_j)}\right) \\
& - 2\hat{y}_s\left(\sum_{j=1,j\neq s}^{m}\frac{k^T(\hat{p}_j)\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)\Bigg)\lambda(\hat{p}_s) - 2b\left(\sum_{j=1,j\neq s}^{m}\frac{k^T(\hat{p}_j)}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s) \\
& + a^2\Bigg(\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)^2\left(\sum_{j=1,j\neq s}^{m}\frac{1}{\hat{z}(\hat{p}_j)}\right)+2\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)\left(\sum_{j=1,j\neq s}^{m}\frac{(\hat{x}_j{}^2+\hat{y}_j{}^2)}{\hat{z}(\hat{p}_j)}\right) \\
& - 4\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)\hat{x}_s\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{x}_j}{\hat{z}(\hat{p}_j)}\right)-4\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)\hat{y}_s\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)+\left(\sum_{j=1,j\neq s}^{m}\frac{(\hat{x}_j{}^2+\hat{y}_j{}^2)^2}{\hat{z}(\hat{p}_j)}\right) \\
& - 4\hat{x}_s\left(\sum_{j=1,j\neq s}^{m}\frac{(\hat{x}_j{}^2+\hat{y}_j{}^2)\hat{x}_j}{\hat{z}(\hat{p}_j)}\right)-4\hat{y}_s\left(\sum_{j=1,j\neq s}^{m}\frac{(\hat{x}_j{}^2+\hat{y}_j{}^2)\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)+4\hat{x}_s{}^2\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{x}_j{}^2}{\hat{z}(\hat{p}_j)}\right) \\
& + 8\hat{x}_s\hat{y}_s\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{x}_j\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)+4\hat{y}_s{}^2\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{y}_j{}^2}{\hat{z}(\hat{p}_j)}\right)\Bigg)+b^2\sum_{j=1,j\neq s}^{m}\frac{1}{\hat{z}(\hat{p}_j)} \\
& + 2ab\Bigg(\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)\left(\sum_{j=1,j\neq s}^{m}\frac{1}{\hat{z}(\hat{p}_j)}\right)+\left(\sum_{j=1,j\neq s}^{m}\frac{(\hat{x}_j{}^2+\hat{y}_j{}^2)}{\hat{z}(\hat{p}_j)}\right)-2\hat{x}_s\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{x}_j}{\hat{z}(\hat{p}_j)}\right) \\
& - 2\hat{y}_s\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)\Bigg)\Bigg)+\frac{\hat{\sigma}^2(\hat{p}_s)}{\hat{z}(\hat{p}_S)}
\end{aligned}
\tag{3.58}
$$

$$
\begin{aligned}
imp_{SM}(\hat{p}_s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)}\Bigg(& \lambda^T(\hat{p}_s)\left(\sum_{j=1,j\neq s}^{m}\frac{k(\hat{p}_j)k^T(\hat{p}_j)}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s)-\left(2a\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)+2b\right)\left(\sum_{j=1,j\neq s}^{m}\frac{k^T(\hat{p}_j)}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s) \\
& - 2a\left(\sum_{j=1,j\neq s}^{m}\frac{k^T(\hat{p}_j)(\hat{x}_j{}^2+\hat{y}_j{}^2)}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s)+4a\hat{x}_s\left(\sum_{j=1,j\neq s}^{m}\frac{k^T(\hat{p}_j)\hat{x}_j}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s) \\
& + 4a\hat{y}_s\left(\sum_{j=1,j\neq s}^{m}\frac{k^T(\hat{p}_j)\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s)+\left(a^2\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)^2+2ab\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)+b^2\right)\left(\sum_{j=1,j\neq s}^{m}\frac{1}{\hat{z}(\hat{p}_j)}\right) \\
& + \left(2a^2\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)+2ab\right)\left(\sum_{j=1,j\neq s}^{m}\frac{(\hat{x}_j{}^2+\hat{y}_j{}^2)}{\hat{z}(\hat{p}_j)}\right)-\left(4a^2\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)\hat{x}_s+4ab\hat{x}_s\right)\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{x}_j}{\hat{z}(\hat{p}_j)}\right) \\
& - \left(4a^2\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)\hat{y}_s+4ab\hat{y}_s\right)\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)+a^2\left(\sum_{j=1,j\neq s}^{m}\frac{(\hat{x}_j{}^2+\hat{y}_j{}^2)^2}{\hat{z}(\hat{p}_j)}\right) \\
& - 4a^2\hat{x}_s\left(\sum_{j=1,j\neq s}^{m}\frac{(\hat{x}_j{}^2+\hat{y}_j{}^2)\hat{x}_j}{\hat{z}(\hat{p}_j)}\right)-4a^2\hat{y}_s\left(\sum_{j=1,j\neq s}^{m}\frac{(\hat{x}_j{}^2+\hat{y}_j{}^2)\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)+4a^2\hat{x}_s{}^2\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{x}_j{}^2}{\hat{z}(\hat{p}_j)}\right) \\
& + 8a^2\hat{x}_s\hat{y}_s\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{x}_j\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)+4a^2\hat{y}_s{}^2\left(\sum_{j=1,j\neq s}^{m}\frac{\hat{y}_j{}^2}{\hat{z}(\hat{p}_j)}\right)\Bigg)+\frac{\hat{\sigma}^2(\hat{p}_s)}{\hat{z}(\hat{p}_S)}
\end{aligned}
\tag{3.59}
$$

Since it is desirable to only calculate the summations in the equation once for each triangle, the $j \neq s$ criterion needs to be eliminated. To remove this criterion, the summation will be adjusted to remove the $j \neq s$ restriction, and then to compensate for the additional $j = s$ term that was added into the summation, it will be subtracted outside of the summation. Furthermore, as $U\left(h(\hat{p}_s, \hat{p}_j)\right)$ has been removed from the summation present in eq. (3.59), $j \neq s$ restriction adds an additional b to the semivariance that then needs to be removed, since $\gamma(0) = 0$. Subtracting the resulting $j = s$ term while removing the $j \neq s$ restriction from eq. (3.59) will produce the final linear equation in which all summations over *j* can be calculated first, followed by the calculations for each point *s*.

Determining the values of the $j = s$ term is performed as follows:

$$Term(j = s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)}\left(\frac{(k^T(\hat{p}_s)\lambda(\hat{p}_s) - ah^2(\hat{p}_s, \hat{p}_s) - b)^2}{\hat{z}(\hat{p}_s)}\right) \tag{3.60}$$

Substituting in $k^T(\hat{p}_s)\lambda(\hat{p}_s) = \hat{\sigma}^2(\hat{p}_s)$, and $h^2(\hat{p}_s, \hat{p}_s) = 0$:

$$Term(j = s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)}\left(\frac{(\hat{\sigma}^2(\hat{p}_s) - b)^2}{\hat{z}(\hat{p}_s)}\right) \tag{3.61}$$

$$Term(j = s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)}\left(\frac{\hat{\sigma}^4(\hat{p}_s) - 2b\hat{\sigma}^2(\hat{p}_s) + b^2}{\hat{z}(\hat{p}_s)}\right) \tag{3.62}$$

$$Term(j = s) = \frac{\hat{\sigma}^2(\hat{p}_s)}{\hat{z}(\hat{p}_s)} - \frac{2b}{\hat{z}(\hat{p}_s)} + \frac{b^2}{\hat{\sigma}^2(\hat{p}_s)\hat{z}(\hat{p}_s)} \tag{3.63}$$

Therefore the final efficient improvement equation becomes:

$$
\begin{aligned}
imp_{SM}(\hat{p}_s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)}\Bigg(& \lambda^T(\hat{p}_s)\left(\sum_{j=1}^{m}\frac{k(\hat{p}_j)k^T(\hat{p}_j)}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s) - \left(2a\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)+2b\right)\left(\sum_{j=1}^{m}\frac{k^T(\hat{p}_j)}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s) \\
& - 2a\left(\sum_{j=1}^{m}\frac{k^T(\hat{p}_j)\left(\hat{x}_j{}^2+\hat{y}_j{}^2\right)}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s) + 4a\hat{x}_s\left(\sum_{j=1}^{m}\frac{k^T(\hat{p}_j)\hat{x}_j}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s) + 4a\hat{y}_s\left(\sum_{j=1}^{m}\frac{k^T(\hat{p}_j)\hat{y}_j}{\hat{z}(\hat{p}_j)}\right)\lambda(\hat{p}_s) \\
& + \left(a^2\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)^2+2ab\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)+b^2\right)\left(\sum_{j=1}^{m}\frac{1}{\hat{z}(\hat{p}_j)}\right) + \left(2a^2\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)+2ab\right)\left(\sum_{j=1}^{m}\frac{\left(\hat{x}_j{}^2+\hat{y}_j{}^2\right)}{\hat{z}(\hat{p}_j)}\right) \\
& - \left(4a^2\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)\hat{x}_s+4ab\hat{x}_s\right)\left(\sum_{j=1}^{m}\frac{\hat{x}_j}{\hat{z}(\hat{p}_j)}\right) - \left(4a^2\left(\hat{x}_s{}^2+\hat{y}_s{}^2\right)\hat{y}_s+4ab\hat{y}_s\right)\left(\sum_{j=1}^{m}\frac{\hat{y}_j}{\hat{z}(\hat{p}_j)}\right) \\
& + a^2\left(\sum_{j=1}^{m}\frac{\left(\hat{x}_j{}^2+\hat{y}_j{}^2\right)^2}{\hat{z}(\hat{p}_j)}\right) - 4a^2\hat{x}_s\left(\sum_{j=1}^{m}\frac{\left(\hat{x}_j{}^2+\hat{y}_j{}^2\right)\hat{x}_j}{\hat{z}(\hat{p}_j)}\right) - 4a^2\hat{y}_s\left(\sum_{j=1}^{m}\frac{\left(\hat{x}_j{}^2+\hat{y}_j{}^2\right)\hat{y}_j}{\hat{z}(\hat{p}_j)}\right) \\
& + 4a^2\hat{x}_s{}^2\left(\sum_{j=1}^{m}\frac{\hat{x}_j{}^2}{\hat{z}(\hat{p}_j)}\right) + 8a^2\hat{x}_s\hat{y}_s\left(\sum_{j=1}^{m}\frac{\hat{x}_j\hat{y}_j}{\hat{z}(\hat{p}_j)}\right) + 4a^2\hat{y}_s{}^2\left(\sum_{j=1}^{m}\frac{\hat{y}_j{}^2}{\hat{z}(\hat{p}_j)}\right)\Bigg) + \frac{2b}{\hat{z}(\hat{p}_s)} - \frac{b^2}{\hat{\sigma}^2(\hat{p}_s)\hat{z}(\hat{p}_s)}
\end{aligned}
\tag{3.64}
$$

As can be seen from eq. (3.64), the summations can be done over all m unknown points once. These summations are then used at each local data point, $\hat{p}_s$, for the local calculation of estimated improvement. This is due to the distance squared semivariogram approximation made in Section 3.1.1, and incorporated in eq. (3.54) in order to develop the surface mesh improvement measure, otherwise the *s* and *j* terms could not be separated for efficient linear time calculation with respect to the number of unknown points (i.e. the distance calculation would need to be performed for each set of unknown points, yielding a parabolic time calculation with respect to the number of unknown points). Note that when calculating values for the Kriging system itself, the semivariogram where $w = 2 - \varepsilon$ is used, and not the approximation where $w = 2$, in order to prevent the situation of singular matrices caused by an impermissible semivariogram.

3.1.3 Surface Mesh Improvement Algorithm Summary

The original surface mesh based improvement algorithm uses the VMR as a basis for formulating the improvement measure, which was motivated by the depth estimate and variance produced by Kriging. Using these formulations as a starting point, the improvement measure is developed by expanding upon the Ordinary Kriging equations by the addition of another point into the existing system, and relating that new system with the additional point to the original system. Furthermore, the improvement for each point is defined as the sum of the differences between the estimated VMR caused by the simulated insertion of a point at any location and the current VMR at the same location over all locations which are contained

within the local region into which the point will be inserted. As the local regions are defined by triangles created by Delaunay triangulation, only triangles which have been changed by the insertion of a new point need to have their improvement values recalculated. This prevents the expensive task of recalculating the improvement over the entire improvement map each time a point is added. This process is illustrated in Figure 3.5. The produced improvement map defines the estimated improvement that an acquisition of one extra range measurement at the particular location in the improvement map will make on the knowledge of the scene.

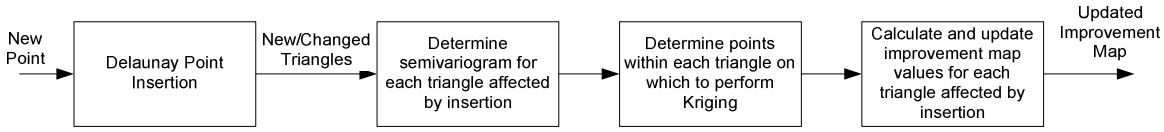


Figure 3.5 - Surface mesh improvement algorithm diagram

3.1.4 Simplified Example

This subsection presents a simplified example showing the improvement map generation process. Figure 3.6 illustrates a configuration containing 6 known points, p_i, where p_1, p_2, and p_3 form a triangle within which the improvement measure will be calculated, and p_4, p_5, and p_6 form the other 3 points from which the semivariogram model will be determined. Table 3.1 contains the locations $(\theta_i,\ \varphi_i)$ for each p_i from a given viewpoint, corresponding to the range sensor location, and the corresponding measured depths, $\rho(p_i)$ using the angular projection coordinates as defined in Appendix A. This will result in the remapping of the variables $x \to c_0 \to \theta$, $y \to c_1 \to \varphi$, and $z \to c_2 \to \rho$ in the surface mesh improvement measure defined in eq. (3.64). There are three locations within the triangle formed by the measured points, p_1, p_2, and p_3, which are identified as $\hat{p}_1$, $\hat{p}_2$, and $\hat{p}_3$, with the locations $\left(\hat{\theta}_j,\ \hat{\varphi}_j\right)$ being identified in Table 3.2. These interior triangle points are in practice determined by a raster triangle filling algorithm, such as the digital differential analyser (DDA) algorithm [81]. Finally, the semivariogram model will use $\varepsilon = 0.00001$ as this value was found to work well in practice, which means that $w = 1.99999$.

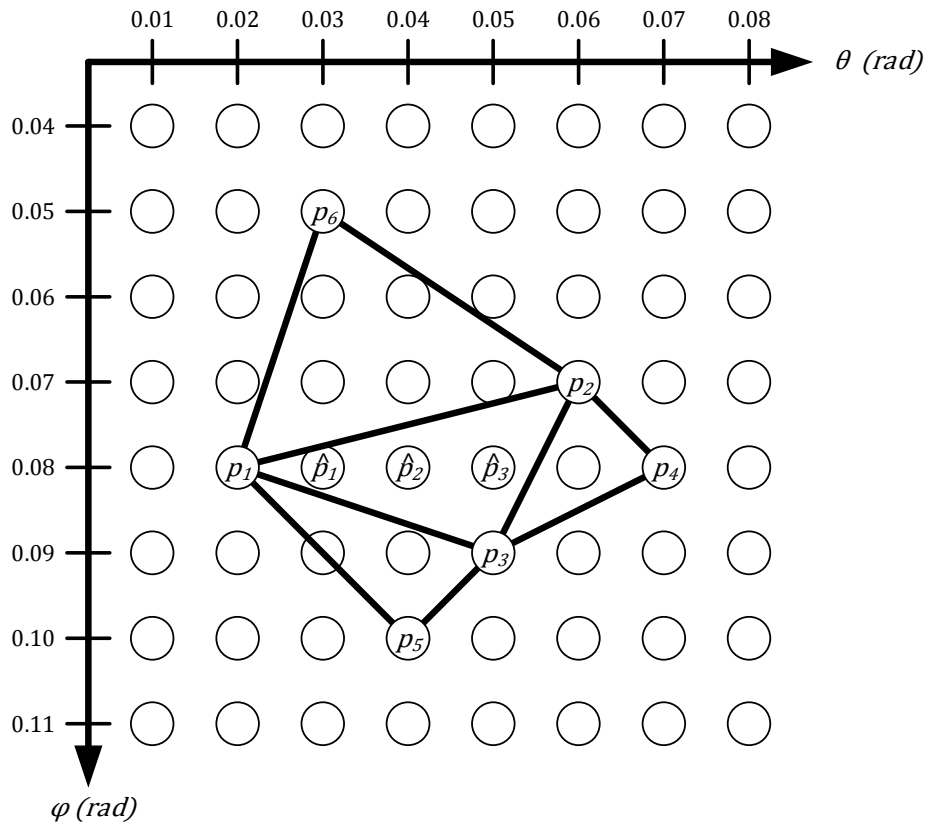


Figure 3.6 - Simplified example for the application of the surface mesh improvement measure

i	p_i	$\rho(p_i)$
1	(0.02, 0.08)	5.0
2	(0.06, 0.07)	5.5
3	(0.05, 0.09)	5.2
4	(0.07, 0.08)	6.0
5	(0.04, 0.10)	5.3
6	(0.03, 0.05)	4.9

Table 3.1 - p_i and $\rho(p_i)$ for each i

j	$\hat{p}_j$
1	(0.03, 0.08)
2	(0.04, 0.08)
3	(0.05, 0.08)

Table 3.2 - $\hat{p}_j$ for each j

The first step for estimating the improvement is to determine the corresponding distance/semivariance pairs of known points as stated in Section 3.1.1. This is performed in Table 3.3. The distance between pairs of points is calculated as in eq. (3.3), where $c_0 = \theta$ and $c_1 = \varphi$. After the corresponding pairs are determined, the parameters a and b from the semivariogram model are calculated by least squares fitting (eq. (3.5)) of the model to the empirical data in Table 3.3.

Pair #	First Point ($\boldsymbol{p_i}$)	Second Point ($\boldsymbol{p_j}$)	Distance ($\boldsymbol{h(p_i, p_j)}$)	Semivariance
1	$\boldsymbol{p_1}$	$\boldsymbol{p_2}$	0.041231	0.125
2	$\boldsymbol{p_1}$	$\boldsymbol{p_3}$	0.031623	0.020
3	$\boldsymbol{p_1}$	$\boldsymbol{p_4}$	0.050000	0.500
4	$\boldsymbol{p_1}$	$\boldsymbol{p_5}$	0.028284	0.045
5	$\boldsymbol{p_1}$	$\boldsymbol{p_6}$	0.031623	0.005
6	$\boldsymbol{p_2}$	$\boldsymbol{p_3}$	0.022361	0.045
7	$\boldsymbol{p_2}$	$\boldsymbol{p_4}$	0.014142	0.125
8	$\boldsymbol{p_2}$	$\boldsymbol{p_5}$	0.036056	0.020
9	$\boldsymbol{p_2}$	$\boldsymbol{p_6}$	0.036056	0.180
10	$\boldsymbol{p_3}$	$\boldsymbol{p_4}$	0.022361	0.320
11	$\boldsymbol{p_3}$	$\boldsymbol{p_5}$	0.014142	0.005
12	$\boldsymbol{p_3}$	$\boldsymbol{p_6}$	0.044721	0.045
13	$\boldsymbol{p_4}$	$\boldsymbol{p_5}$	0.036056	0.245
14	$\boldsymbol{p_4}$	$\boldsymbol{p_6}$	0.050000	0.605
15	$\boldsymbol{p_5}$	$\boldsymbol{p_6}$	0.050990	0.080

Table 3.3 - Corresponding distance/semivariance pairs

$$sn = 15$$
$$sh^{2w} = 3.4442 \times 10^{-5}$$
$$sh^{w} = 1.9401 \times 10^{-2}$$
$$s\gamma = 2.3650$$
$$s\gamma h^{w} = 4.1211 \times 10^{-3}$$

$$\begin{bmatrix} b \\ a \end{bmatrix} = \frac{1}{(15)(3.4442 \times 10^{-5}) - (1.9401 \times 10^{-2})^2} \begin{bmatrix} 3.4442 \times 10^{-5} & -1.9401 \times 10^{-2} \\ -1.9401 \times 10^{-2} & 15 \end{bmatrix} \begin{bmatrix} 2.3650 \\ 4.1211 \times 10^{-3} \end{bmatrix} = \begin{bmatrix} 1.0718 \times 10^{-2} \\ 1.1362 \times 10^{2} \end{bmatrix} \quad (3.65)$$

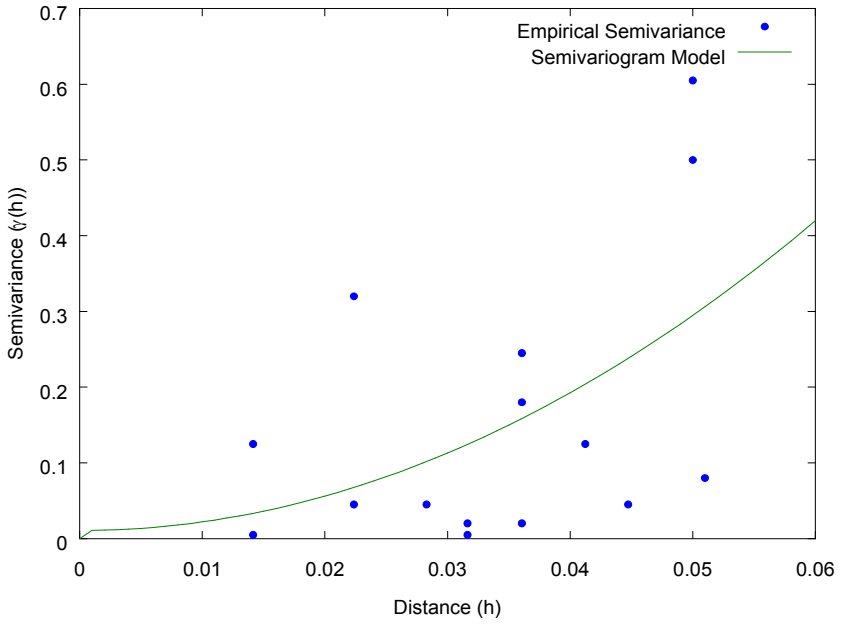


Figure 3.7 - Plot of empirical semivariance and the semivariogram model

Eq. (3.65) shows the calculation of the parameters a and b from the semivariogram model which corresponds to the example triangle, and Figure 3.7 shows the empirical semivariance plotted against the semivariogram model with the parameters obtained. The resulting semivariogram model, using a and b from eq. (3.65) is put into evidence in eq. (3.66).

$$\gamma(h) = 1.1362 \times 10^{2} h^{1.99999} + 1.0718 \times 10^{-2} U(h) \tag{3.66}$$

The application of eq. (3.66) in two scenarios is shown below, that for the case of $\gamma\big(h(p_6, p_2)\big)$ and for the case of $\gamma\big(h(p_1, \hat{p}_3)\big)$. The first step is to calculate the respective distances, as demonstrated in eq. (3.67) and eq. (3.68).

$$h(p_6, p_2) = \sqrt{(0.03 - 0.06)^2 + (0.05 - 0.07)^2} = 0.036056 \tag{3.67}$$

$$h(p_1, \hat{p}_3) = \sqrt{(0.02 - 0.05)^2 + (0.08 - 0.08)^2} = 0.030000 \tag{3.68}$$

After the distances are calculated, the values for $\gamma\big(h(p_6, p_2)\big)$ and $\gamma\big(h(p_1, \hat{p}_3)\big)$ are calculated respectively in eq. (3.69) and eq. (3.70).

$$\begin{aligned} \gamma\big(h(p_6, p_2)\big) &= 1.1362 \times 10^{2} (0.036056)^{1.99999} + 1.0718 \times 10^{-2} U(0.036056) \\ &= 0.15842 \end{aligned} \tag{3.69}$$

$$\begin{aligned} \gamma\big(h(p_1, \hat{p}_3)\big) &= 1.1362 \times 10^{2} (0.030000)^{1.99999} + 1.0718 \times 10^{-2} U(0.030000) \\ &= 0.11298 \end{aligned} \tag{3.70}$$

Now that the semivariogram model is determined, and the locations of the points for which the improvement calculation is to be performed are known; the calculations for the improvement map can be performed. First, the determination of the Kriging system is achieved using eq. (3.12) to eq. (3.17) and the semivariogram model in eq. (3.66). The resulting Kriging system is shown in eq. (3.71) to eq. (3.76).

$$K = \begin{bmatrix} \gamma\big(h(p_1,p_1)\big) & \gamma\big(h(p_1,p_2)\big) & \cdots & \gamma\big(h(p_1,p_6)\big) & 1 \\ \gamma\big(h(p_2,p_1)\big) & \gamma\big(h(p_2,p_2)\big) & \cdots & \gamma\big(h(p_2,p_6)\big) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma\big(h(p_6,p_1)\big) & \gamma\big(h(p_6,p_2)\big) & \cdots & \gamma\big(h(p_6,p_6)\big) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0.20387 & 0.12434 & 0.29477 & 0.10161 & 0.12434 & 1 \\ 0.20387 & 0 & 0.06753 & 0.03344 & 0.15842 & 0.15842 & 1 \\ 0.12434 & 0.06753 & 0 & 0.06753 & 0.03344 & 0.23796 & 1 \\ 0.29477 & 0.03344 & 0.06753 & 0 & 0.15842 & 0.29477 & 1 \\ 0.10161 & 0.15842 & 0.03344 & 0.15842 & 0 & 0.30613 & 1 \\ 0.12434 & 0.15842 & 0.23796 & 0.29477 & 0.30613 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{3.71}$$

$$k(\hat{p}_j) = \begin{bmatrix} \gamma\left(h(p_1,\hat{p}_j)\right) \\ \gamma\left(h(p_2,\hat{p}_j)\right) \\ \vdots \\ \gamma\left(h(p_6,\hat{p}_j)\right) \\ 1 \end{bmatrix} \rightarrow k(\hat{p}_1) = \begin{bmatrix} 0.02208 \\ 0.12434 \\ 0.06753 \\ 0.19251 \\ 0.06753 \\ 0.11298 \\ 1 \end{bmatrix}, \quad k(\hat{p}_2) = \begin{bmatrix} 0.05617 \\ 0.06753 \\ 0.03344 \\ 0.11298 \\ 0.05617 \\ 0.12434 \\ 1 \end{bmatrix}, \quad k(\hat{p}_3) = \begin{bmatrix} 0.11298 \\ 0.03344 \\ 0.02208 \\ 0.05617 \\ 0.06753 \\ 0.15842 \\ 1 \end{bmatrix} \tag{3.72}$$

$$Z = \begin{bmatrix} \rho(p_1) \\ \rho(p_2) \\ \vdots \\ \rho(p_6) \\ 0 \end{bmatrix} = \begin{bmatrix} 5.0 \\ 5.5 \\ 5.2 \\ 6.0 \\ 5.3 \\ 4.9 \\ 0 \end{bmatrix} \tag{3.73}$$

$$\lambda(\hat{p}_j) = K^{-1}k(\hat{p}_j) \rightarrow \quad \lambda(\hat{p}_1) = \begin{bmatrix} 0.38826 \\ 0.01533 \\ 0.15246 \\ -0.04654 \\ 0.26450 \\ 0.22599 \\ -0.04126 \end{bmatrix}, \quad \lambda(\hat{p}_2) = \begin{bmatrix} 0.24435 \\ 0.10849 \\ 0.17010 \\ 0.09424 \\ 0.21589 \\ 0.16694 \\ -0.05757 \end{bmatrix}, \quad \lambda(\hat{p}_3) = \begin{bmatrix} 0.10045 \\ 0.20165 \\ 0.18772 \\ 0.23503 \\ 0.16727 \\ 0.10789 \\ -0.05116 \end{bmatrix} \tag{3.74}$$

$$\hat{\rho}(\hat{p}_j) = \lambda^T(\hat{p}_j)Z \rightarrow \begin{matrix} \hat{\rho}(\hat{p}_1) = 5.0484 \\ \hat{\rho}(\hat{p}_2) = 5.2306 \\ \hat{\rho}(\hat{p}_3) = 5.4128 \end{matrix} \tag{3.75}$$

$$\hat{\sigma}^2(\hat{p}_j) = \lambda^T(\hat{p}_j)k(\hat{p}_j) \rightarrow \begin{matrix} \hat{\sigma}^2(\hat{p}_1) = 0.01395 \\ \hat{\sigma}^2(\hat{p}_2) = 0.01269 \\ \hat{\sigma}^2(\hat{p}_3) = 0.01266 \end{matrix} \tag{3.76}$$

After the Kriging system has been determined, the surface mesh improvement equation, eq. (3.64), can now be applied to calculate the improvement for the three unknown locations, $\hat{p}_1$, $\hat{p}_2$, and $\hat{p}_3$. To do this efficiently, the equation will be broken down such that the common elements for all three unknown points, $\hat{p}_1$, $\hat{p}_2$, and $\hat{p}_3$, can be calculated only once (see Table 3.4), and then by using these common elements the final improvement values can be determined (eq. (3.77)).

$$
\begin{aligned}
imp_{SM}(\hat{p}_s) = \frac{1}{\hat{\sigma}^2(\hat{p}_s)}\Big(&\lambda^T(\hat{p}_s)(\mathrm{S1})\lambda(\hat{p}_s) - \big(2a\big(\hat{x}_s{}^2 + \hat{y}_s{}^2\big) + 2b\big)(\mathrm{S2})\lambda(\hat{p}_s) - 2a(\mathrm{S3})\lambda(\hat{p}_s) + 4a\hat{x}_s(\mathrm{S4})\lambda(\hat{p}_s) \\
&+ 4a\hat{y}_s(\mathrm{S5})\lambda(\hat{p}_s) + \Big(a^2\big(\hat{x}_s{}^2 + \hat{y}_s{}^2\big)^2 + 2ab\big(\hat{x}_s{}^2 + \hat{y}_s{}^2\big) + b^2\Big)(\mathrm{S6}) + \big(2a^2\big(\hat{x}_s{}^2 + \hat{y}_s{}^2\big) + 2ab\big)(\mathrm{S7}) \\
&- \big(4a^2\big(\hat{x}_s{}^2 + \hat{y}_s{}^2\big)\hat{x}_s + 4ab\hat{x}_s\big)(\mathrm{S8}) - \big(4a^2\big(\hat{x}_s{}^2 + \hat{y}_s{}^2\big)\hat{y}_s + 4ab\hat{y}_s\big)(\mathrm{S9}) + a^2(\mathrm{S10}) - 4a^2\hat{x}_s(\mathrm{S11}) \\
&- 4a^2\hat{y}_s(\mathrm{S12}) + 4a^2\hat{x}_s{}^2(\mathrm{S13}) + 8a^2\hat{x}_s\hat{y}_s(\mathrm{S14}) + 4a^2\hat{y}_s{}^2(\mathrm{S15})\Big) + \frac{2b}{\hat{\rho}(\hat{p}_s)} - \frac{b^2}{\hat{\sigma}^2(\hat{p}_s)\hat{\rho}(\hat{p}_s)}
\end{aligned}
\tag{3.77}
$$

$$
\begin{aligned}
imp_{SM}(\hat{p}_1) &= 2.860 \times 10^{-3} \\
imp_{SM}(\hat{p}_2) &= 2.550 \times 10^{-3} \\
imp_{SM}(\hat{p}_3) &= 2.410 \times 10^{-3}
\end{aligned}
$$

The results in eq. (3.77) show that $\hat{p}_1$ has the most estimated potential improvement out of the three points, $\hat{p}_3$ has the least estimated improvement, while $\hat{p}_2$ is in between the other two. By observation of the layout of the points in Figure 3.6, this makes sense from a purely geometric situation, as $\hat{p}_3$ is closer to more known measured points than the other two, while $\hat{p}_1$ is only relatively close to a single known measured point.

Label	Summation	Result
S1	$\sum_{j=1}^{3} \frac{k(\hat{p}_j)k^T(\hat{p}_j)}{\hat{\rho}(\hat{p}_j)}$	$\begin{bmatrix} 0.00306 & 0.00197 & 0.00112 & 0.00323 & 0.00231 & 0.00514 & 0.03598 \\ 0.00197 & 0.00414 & 0.00223 & 0.00655 & 0.00281 & 0.00537 & 0.04372 \\ 0.00112 & 0.00223 & 0.00121 & 0.00353 & 0.00154 & 0.00295 & 0.02385 \\ 0.00322 & 0.00655 & 0.00353 & 0.01036 & 0.00449 & 0.00864 & 0.07011 \\ 0.00231 & 0.00281 & 0.00154 & 0.00449 & 0.00235 & 0.00482 & 0.03659 \\ 0.00514 & 0.00537 & 0.00295 & 0.00864 & 0.00482 & 0.01012 & 0.07542 \\ 0.03598 & 0.04372 & 0.02385 & 0.07011 & 0.03659 & 0.07542 & 0.57402 \end{bmatrix}$
S2	$\sum_{j=1}^{3} \frac{k^T(\hat{p}_j)}{\hat{\rho}(\hat{p}_j)}$	$[0.03598 \quad 0.04372 \quad 0.02385 \quad 0.07011 \quad 0.03659 \quad 0.07542 \quad 0.57402]$
S3	$\sum_{j=1}^{3} \frac{k^T(\hat{p}_j)(\hat{x}_j{}^2 + \hat{y}_j{}^2)}{\hat{\rho}(\hat{p}_j)}$	$[3.036 \quad 3.381 \quad 1.851 \quad 5.435 \quad 2.946 \quad 6.140 \quad 46.20] \times 10^{-4}$
S4	$\sum_{j=1}^{3} \frac{k^T(\hat{p}_j)\hat{x}_j}{\hat{\rho}(\hat{p}_j)}$	$[1.604 \quad 1.564 \quad 0.861 \quad 2.527 \quad 1.455 \quad 3.086 \quad 22.83] \times 10^{-3}$
S5	$\sum_{j=1}^{3} \frac{k^T(\hat{p}_j)\hat{y}_j}{\hat{\rho}(\hat{p}_j)}$	$[2.889 \quad 3.498 \quad 1.908 \quad 5.609 \quad 2.927 \quad 6.034 \quad 45.92] \times 10^{-3}$
S6	$\sum_{j=1}^{3} \frac{1}{\hat{\rho}(\hat{p}_j)}$	0.5740
S7	$\sum_{j=1}^{3} \frac{(\hat{x}_j{}^2 + \hat{y}_j{}^2)}{\hat{\rho}(\hat{p}_j)}$	4.620×10^{-3}
S8	$\sum_{j=1}^{3} \frac{\hat{x}_j}{\hat{\rho}(\hat{p}_j)}$	2.283×10^{-2}
S9	$\sum_{j=1}^{3} \frac{\hat{y}_j}{\hat{\rho}(\hat{p}_j)}$	4.592×10^{-2}
S10	$\sum_{j=1}^{3} \frac{(\hat{x}_j{}^2 + \hat{y}_j{}^2)^2}{\hat{\rho}(\hat{p}_j)}$	3.743×10^{-5}
S11	$\sum_{j=1}^{3} \frac{(\hat{x}_j{}^2 + \hat{y}_j{}^2)\hat{x}_j}{\hat{\rho}(\hat{p}_j)}$	1.868×10^{-4}
S12	$\sum_{j=1}^{3} \frac{(\hat{x}_j{}^2 + \hat{y}_j{}^2)\hat{y}_j}{\hat{\rho}(\hat{p}_j)}$	3.696×10^{-4}
S13	$\sum_{j=1}^{3} \frac{\hat{x}_j{}^2}{\hat{\rho}(\hat{p}_j)}$	9.460×10^{-4}
S14	$\sum_{j=1}^{3} \frac{\hat{x}_j\hat{y}_j}{\hat{\rho}(\hat{p}_j)}$	1.826×10^{-3}
S15	$\sum_{j=1}^{m} \frac{\hat{y}_j{}^2}{\hat{\rho}(\hat{p}_j)}$	3.674×10^{-3}

Table 3.4 - Intermediate summations for improvement calculation

3.2 Probabilistic Occupancy Grid Improvement Measure

The surface mesh improvement measure developed in Section 3.1 focuses on the improvement of the accurate representation of details in the scene, which is not necessarily the goal in all situations, especially in applications when greater knowledge involving the coverage of the scene is also desired, such as detecting relatively small scale features that are not detected by the initial sampling. A probabilistic occupancy grid, as described in Section 2.4.4, on the other hand, can be used to obtain a measure of spatial coverage. This occurs due to the fact that when a single point is inserted into a probabilistic occupancy grid, it tends to only influence the probabilistic volume about the local region, depending on the occupancy probability distribution function (OPDF) of the sensor used. The unmeasured regions that have a 0.5 probability of occupancy translates into regions where additional measurements could be made to improve coverage. Additionally, a probabilistic occupancy grid only provides positional accuracy up to the size of a grid cell, and does not perform any implicit interpolation between acquired points, and as such it is a poor choice to use for capturing scene accuracy. This section develops a novel technique, different than that presented in Section 3.1, to provide an estimate of improvement based upon probabilistic occupancy grids. This measure will provide the necessary information to ensure that coverage is considered in the selective sensing process, and hence will be used to balance the accuracy of representation for surface details that is obtained from surface meshes developed in Section 3.1.

Payeur *et al.* [67] developed a fast approximation of Elfes' [65] work on probabilistic occupancy grids, which yielded a sensor OPDF, $P_s\left(\rho,\theta,\varphi\middle|\bar{\rho},\bar{\theta},\bar{\varphi},\sigma_\rho,\sigma_\theta,\sigma_\varphi\right)$, as shown in eq. (3.78), where ρ,θ,φ correspond to the depth, azimuth, and elevation values in a spherical occupancy grid, $\bar{\rho},\bar{\theta},\bar{\varphi}$ correspond to the measured location using the sensor, and $\sigma_\rho,\sigma_\theta,\sigma_\varphi$ correspond to the standard deviation of the sensor measurements along the respective axes.

$$P_s\left(\rho,\theta,\varphi\middle|\bar{\rho},\bar{\theta},\bar{\varphi},\sigma_\rho,\sigma_\theta,\sigma_\varphi\right) = \frac{1}{2}\left(1+e^{-\left(\frac{2\left((\rho-\bar{\rho})+2\sigma_\rho\right)}{\sigma_\rho}+\frac{(\theta-\bar{\theta})^2}{\sigma_\theta{}^2}+\frac{(\varphi-\bar{\varphi})^2}{\sigma_\varphi{}^2}\right)}\right)^{-1} + \frac{1}{3}e^{-\left(\frac{(\rho-\bar{\rho})^2}{\sigma_\rho{}^2}+\frac{(\theta-\bar{\theta})^2}{\sigma_\theta{}^2}+\frac{(\varphi-\bar{\varphi})^2}{\sigma_\varphi{}^2}\right)} \tag{3.78}$$

The classical Bayesian model is applied to perform data fusion and is formulated as follows:

$$P\left(c^{t+1}(\rho,\theta,\varphi)=OCC\middle|P(c^{t}(\rho,\theta,\varphi)=OCC),P_s(\rho,\theta,\varphi|\bar{\rho},\bar{\theta},\bar{\varphi},\sigma_\rho,\sigma_\theta,\sigma_\varphi)\right)$$
$$=\frac{P(c^{t}(\rho,\theta,\varphi)=OCC)P_s(\rho,\theta,\varphi|\bar{\rho},\bar{\theta},\bar{\varphi},\sigma_\rho,\sigma_\theta,\sigma_\varphi)}{P(c^{t}(\rho,\theta,\varphi)=OCC)P_s(\rho,\theta,\varphi|\bar{\rho},\bar{\theta},\bar{\varphi},\sigma_\rho,\sigma_\theta,\sigma_\varphi)+\left(1-P(c^{t}(\rho,\theta,\varphi)=OCC)\right)\left(1-P_s(\rho,\theta,\varphi|\bar{\rho},\bar{\theta},\bar{\varphi},\sigma_\rho,\sigma_\theta,\sigma_\varphi)\right)} \quad (3.79)$$

Equation (3.79) above describes the probability that the cell, c, at spherical location (ρ,θ,φ) in the occupancy grid is occupied at time $t+1$, based upon its current occupancy at time t, $P(c^{t}(\rho,\theta,\varphi)=OCC)$, and the probability of occupancy based upon a sensor measurement $P_s\left(\rho,\theta,\varphi\middle|\bar{\rho},\bar{\theta},\bar{\varphi},\sigma_\rho,\sigma_\theta,\sigma_\varphi\right)$. As this notation is unwieldy to deal with for mathematical manipulation, the reformulation of eq. (3.79) is desired. Let v_i be the voxel in which the point (ρ,θ,φ) is located, a_k be the acquisition of data at point $(\bar{\rho},\bar{\theta},\bar{\varphi})$ with the sensor standard deviations of $(\sigma_\rho,\sigma_\theta,\sigma_\varphi)$, $P_o^t(v_i)$ be the probability that voxel v_i is occupied at time t, and $P_s(v_i|a_k)$ be the probability that voxel v_i is occupied given a measurement from the sensor with values of a_k, which transforms the Bayesian update equation to be that of eq. (3.80).

$$P_o^{t+1}\left(v_i\middle|P_o^t(v_i),P_s(v_i|a_k)\right)=\frac{P_o^t(v_i)\,P_s(v_i|a_k)}{P_o^t(v_i)\,P_s(v_i|a_k)+(1-P_o^t(v_i)\,)\left(1-P_s(v_i|a_k)\right)} \quad (3.80)$$

Based upon the Bayesian relationship described in eq. (3.80), it is possible to express the estimated future value of occupancy based upon the current value of occupancy, as well as the possible future acquisition at a particular point. To perform this calculation over the entire 3D occupancy grid is computationally expensive, and unneeded since a single point of view is considered here. Within the probabilistic occupancy grid acquired from a single point of view, there will be a surface of highest probability determined by the surface of the closest object in any given direction within the field of view of the range sensor, taking into account that behind any object the volume is occluded from the sensor and its probability of occupancy remains 50% (unknown) by default. This surface of highest probability is created by determining the depth at which the highest probability of occupancy occurs for each azimuth and elevation value represented in the spherical occupancy grid. Figure 3.8 illustrates the creation of the surface of highest probability for a slice of the spherical occupancy grid containing a fixed elevation, and a varying azimuth and depth. Black cell shading represents high occupancy, white cell shading represents low occupancy, or empty, and grey cell shading represents values in between. The numbers represent discrete depth

into the probabilistic occupancy grid. Taking into account this piece of information, the problem is reduced from 3D to 2D, and the computational requirements of the calculation of the Bayesian update in order to determine the potential improvement resulting from acquiring a point of data is reduced to calculations along the surface of highest probability. Furthermore, by limiting the sensor's OPDF region of influence by truncating it to a region within a multiple of the sensor's respective standard deviations along the azimuth and elevation axes, only a limited number of voxels need to have their estimates of improvement updated after the acquisition of another point.

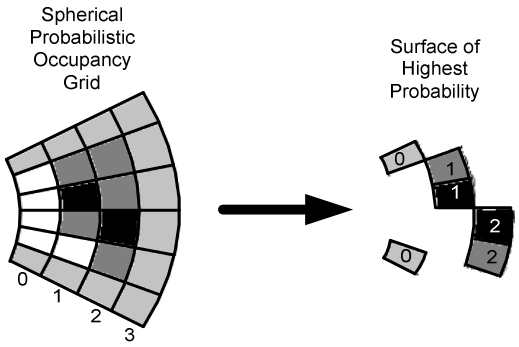


Figure 3.8 - Probabilistic occupancy grid and corresponding surface of highest probability

In order to maximize the knowledge of whether or not space is occupied or empty, and whether acquiring a depth value along a particular ray will maximize the knowledge of the scanned area, a measure of improvement in the knowledge of the probability of occupancy is required. To determine an estimate of improvement of the knowledge of occupancy that a particular point will make if acquired in the future, a simulated acquisition, $\hat{a}_j$, is performed along the ray (θ_j, φ_j), with a depth, ρ_j, corresponding to that of the most probabilistic surface in that specific sensing direction. The current probability of occupancy in the corresponding voxel in spherical coordinates is subtracted from the new probability of occupancy of the same voxel assuming that the simulated acquisition is merged into the occupancy grid. This defines the estimated improvement in the probabilistic occupancy grid (POG), $imp_{POG}\left(v_i \middle| P_o^t(v_i), P_s(v_i|\,\hat{a}_j)\right)$, as follows:

$$imp_{POG}\left(v_i\middle|P_o^t(v_i),P_s(v_i|\hat{a}_j)\right)=P_o^{t+1}\left(v_i\middle|P_o^t(v_i),P_s(v_i|\hat{a}_j)\right)-P_o^t(v_i)$$
$$=\frac{P_o^t(v_i)\,P_s(v_i|\hat{a}_j)}{P_o^t(v_i)\,P_s(v_i|\hat{a}_j)+(1-P_o^t(v_i))\left(1-P_s(v_i|\hat{a}_j)\right)}-P_o^t(v_i) \tag{3.81}$$

The change in the probability of a single voxel when a new acquisition is made is not enough to determine the overall improvement in the confidence of occupancy. Rather the total summation of this improvement happening over all voxels which the acquisition at $\hat{a}_j$ affects, according to the OPDF, must be performed. Additionally, since the OPDF model is based on a Gaussian function where most of the distribution is located close in to the measured point, the OPDF can be truncated such that only voxels within a selected multiple of the sensor model's standard deviations in the parametric definition of the scanning direction (here, the azimuth and elevation are the directions) are used for the calculations. This truncation eliminates cells which would yield very little change to the overall probability and saves a large number of calculations in the process. Furthermore only the voxels currently belonging to the surface of highest probability are used, as defined in eq. (3.82).

$$imp_{POG}(\hat{a}_j)=\sum_{\forall i\,\in\, truncated\ OPDF} imp_{POG}\left(v_i\middle|P_o^t(v_i),P_s(v_i|\hat{a}_j)\right) \tag{3.82}$$

As the distribution of $P_s(v_i|\hat{a}_j)$ is the probability distribution of the possible sensor acquisition along the most probable surface for the purpose of calculating improvement, only the voxels around the region of support provided by the standard deviation of the sensor model need to be used. This effectively makes the improvement calculation a non-linear rectangular filter in 2D on the azimuth and elevation axes. Furthermore, by keeping track of which elements of the surface of highest probabilistic occupancy have changed after the addition of a point, the number of points updated after an acquisition can also be limited to the size of the sensors OPDF, within a multiple of its standard deviation. The process of generating the improvement map is detailed in Figure 3.9.

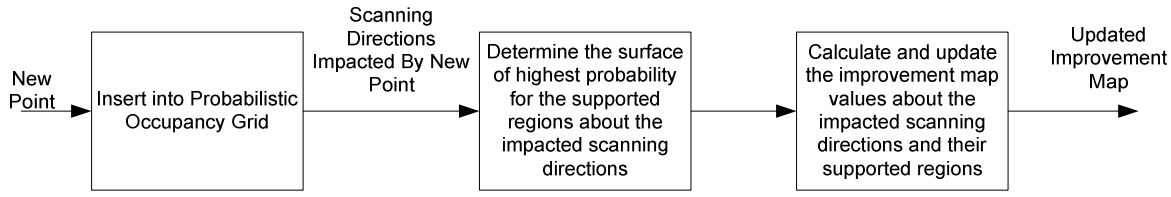


Figure 3.9 - Probabilistic occupancy grid based improvement map algorithm diagram

Beyond asserting the potential improvement on the knowledge about a scene, an interesting characteristic of the proposed improvement measure based on probabilistic occupancy grids is that the resulting cumulative improvement, $imp_{POG}(\hat{a}_j)$, can be negative. Such a negative improvement can be interpreted as uncertainty in the degree of overall improvement in a given region, which can be due to variable sampling densities as well as depth transitions.

Figure 3.10 shows how negative improvement can be created when a point is added to the probabilistic occupancy grid. When such an addition happens, the OPDF leaves a 'hole' of low probability along the projective rays corresponding to the truncated range in the probabilistic occupancy grid until it reaches the measurement depth. The regions around these projective rays still have an unknown probability (0.5), and as such there is no depth measurements to be associated with these surrounding rays (hence the depth is assigned as 0 in this case). Maximum improvement is found where there is no influence by existing points (i.e. locations outside of the supported region covered by the OPDF), as there is the greatest room for change in occupancy. Furthermore where there are existing points, since there is already a higher than 0.5 probability of occupancy, there is less improvement possible in this region. The technique for calculating imp_{POG} assumes that the virtual acquisition occurs along the surface of highest probability of occupancy. On the outside edge of the region of support for the OPDF a location with a large depth value which has a high probability of occupancy is adjacent to a location with a zero depth which has a 0.5 probability of occupancy, between which the OPDF function for the simulated scan determines a probability of occupancy, $P_s(v_i|\hat{a}_j)$, that is close to zero. The negative improvement that results is due to the estimated future probability of occupancy being less than the current probability of occupancy, $P_o^t(v_i)$, that being subtracted by the current probability of occupancy. This effect occurs along the outside edge of the region of influence of the OPDF creates, as illustrated in Figure 3.10. As points are added to the probabilistic occupancy grid, such that their OPDFs overlap, the amount of negative improvement per inserted point decreases.

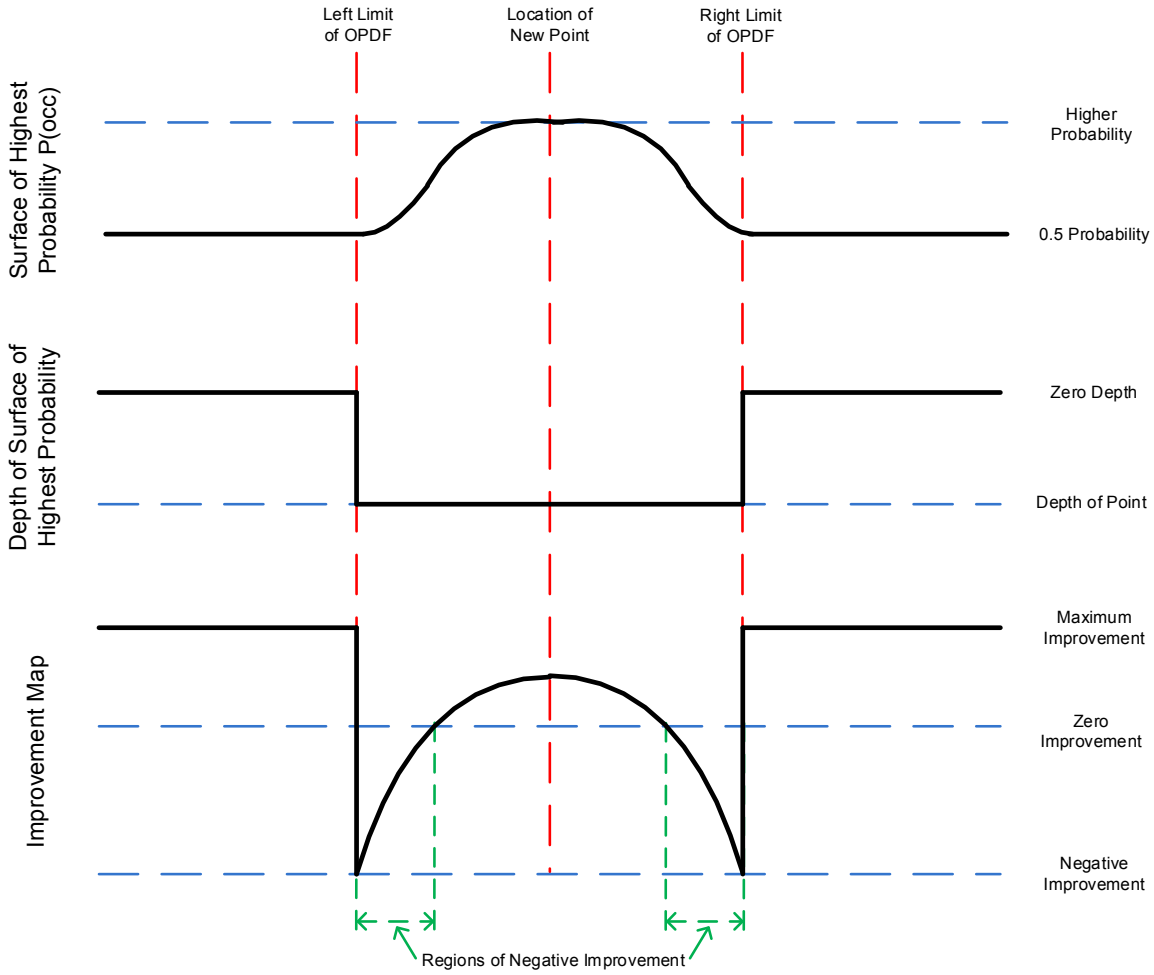


Figure 3.10 - Illustrating the effect adding a point in the probabilistic occupancy grid improvement map

3.2.1 Simplified Example

This section provides a simplified example detailing the creation of the probabilistic occupancy grid and the results of applying the probabilistic occupancy grid based improvement measure. The same configuration of points as what was presented in Section 3.1.4 for the surface mesh improvement measure simplified example is being used, as shown in Figure 3.11.

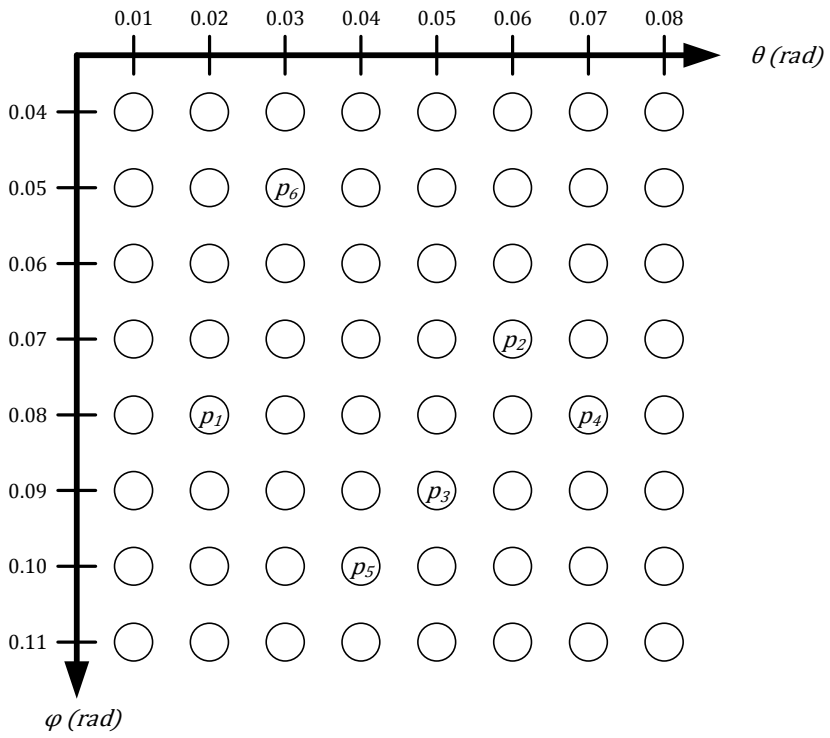


Figure 3.11 - Simplified example used for the application of the probabilistic occupancy grid based improvement measure

In the scenario shown in Figure 3.11, it will be assumed that locations from p_1 through p_6 correspond to measurements already inserted into the probabilistic occupancy grid. The OPDF sensor characteristics are as described in Table 3.6, with the grid discretization as shown in Figure 3.11 (namely 0.01 discretization in both the θ and φ directions). The region of support for the OPDF will be 2 standard deviations, and the depth resolution of the probabilistic occupancy grid will be 0.1 depth unit extending from a depth of 2.0 units to a depth of 7.0 units.

i	p_i	$\rho(p_i)$
1	(0.02, 0.08)	5.0
2	(0.06, 0.07)	5.5
3	(0.05, 0.09)	5.2
4	(0.07, 0.08)	6.0
5	(0.04, 0.10)	5.3
6	(0.03, 0.05)	4.9

Table 3.5 - Measured values inserted into the probabilistic occupancy grid

OPDF Parameter	Value
σ_ρ	0.2
σ_θ	0.01
σ_φ	0.01

Table 3.6 - Example sensor OPDF parameters

Firstly, the probabilistic occupancy grid needs to be calculated for these points, and the surface corresponding to the highest probability in each of the possible scanning directions must be extracted, as illustrated in Figure 3.12 and in Figure 3.13. Note that scanning directions (corresponding to individual (θ, φ) coordinate pairs) which have a probability of occupancy of 0.5 for all depths correspond to directions where there are no measured points, and as such have a depth of 0 associated with them (as shown by the peaks in Figure 3.12). Also note that Figure 3.13 is actually Figure 3.12 flattened and split into two parts, the depth component (Figure 3.13 a)) and the probability of occupancy component mapped in grayscale (Figure 3.13 b)), for easier readability. With the surface corresponding to the highest probability of occupancy being identified (mapped by yellow-red shadings in Figure 3.12 and brighter areas in Figure 3.13 b)), the improvement measure, as defined in eq. (3.82), is used to determine the map of improvement, which is shown in Figure 3.14. Notice that in this improvement map, the region surrounding the known cluster of points contains negative improvement (shown in shades of blue), with small positive improvement in the known cluster of points (shown in shades of yellow), and high improvement outside of the negative improvement region (shown in shades of red). The highest improvement values coincide to locations that would improve coverage of the scene, in other words, directions where there are no measured points.

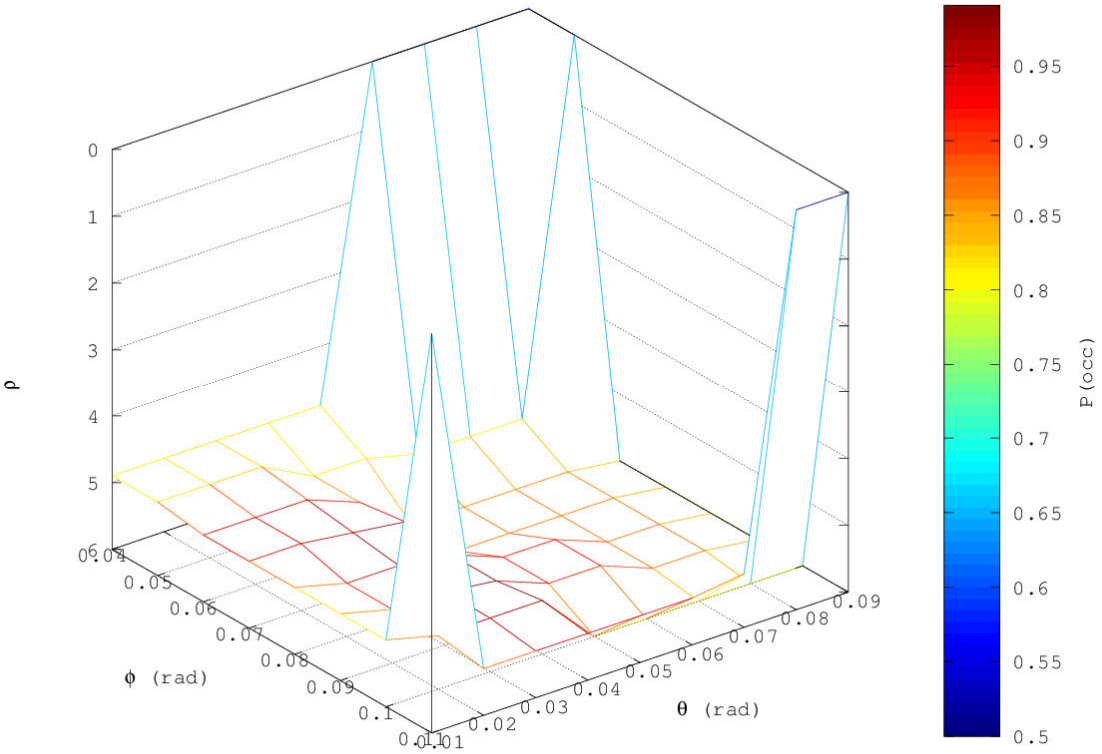
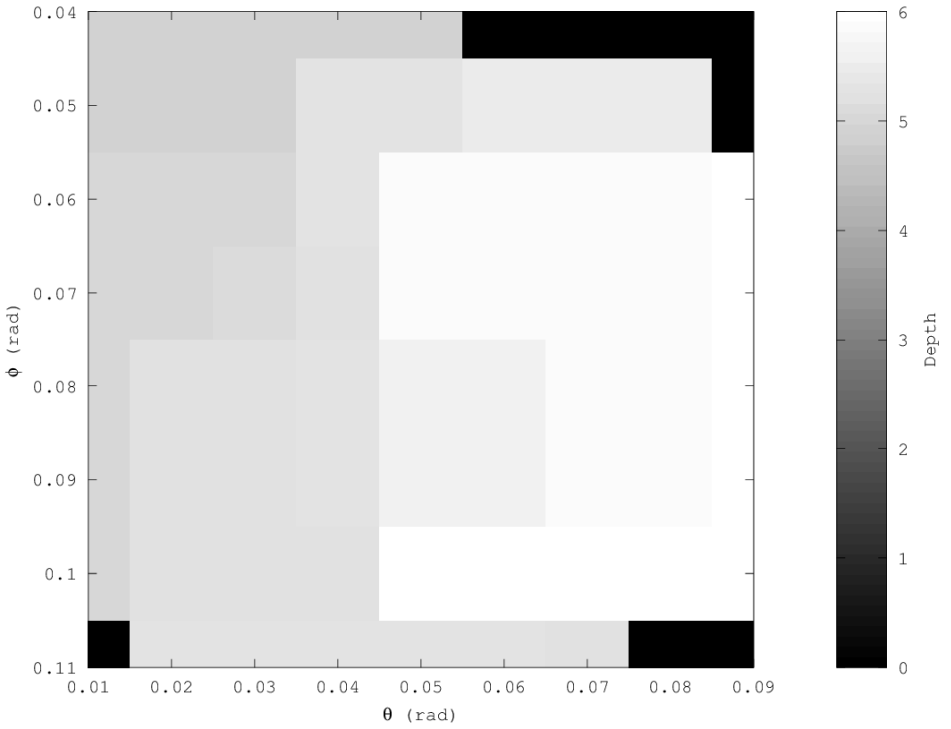
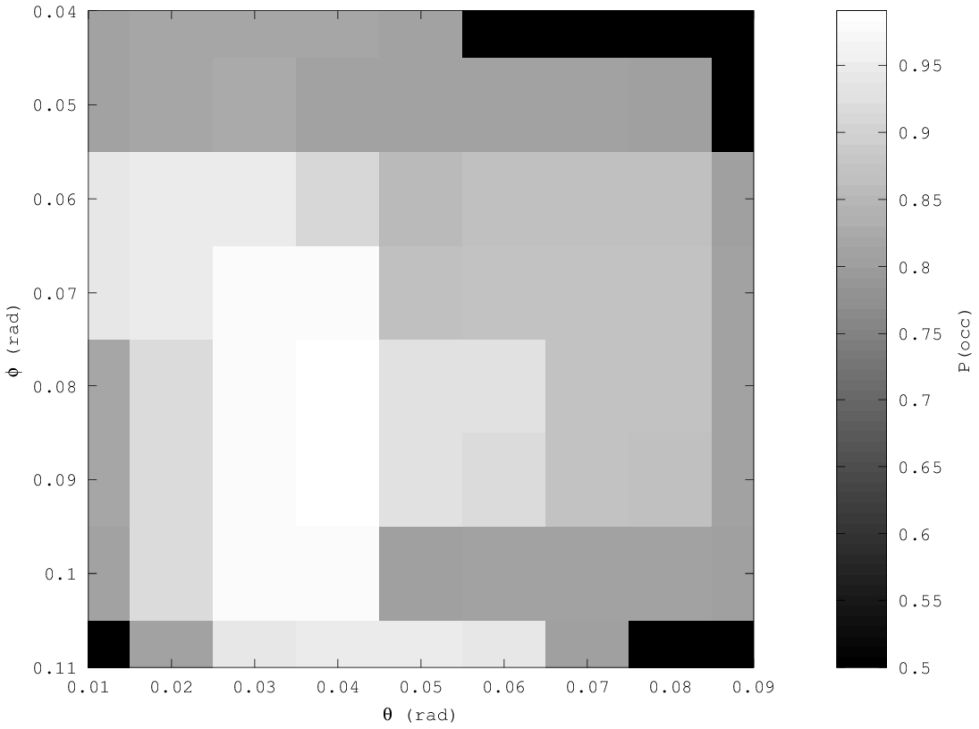


Figure 3.12 - Surface corresponding to the highest probability of occupancy



a)



b)

Figure 3.13 - a) Depth map and b) probability of occupancy map corresponding to the surface of highest probability of occupancy

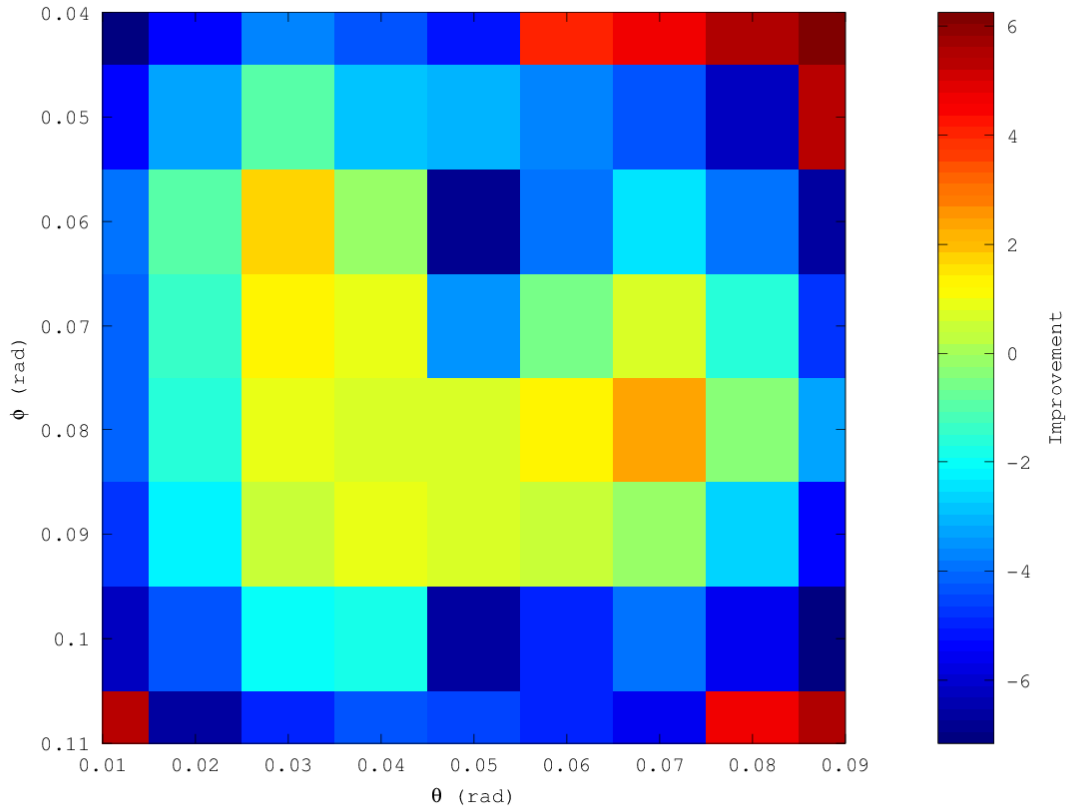


Figure 3.14 - Probabilistic occupancy grid based improvement measure map

3.3 Point Selection Processes

Developing the improvement measures by themselves only provide part of a selective sensing solution. A complete data driven selective sensing algorithm must have a point selection process to choose the next location to acquire based on the provided improvement maps, as is illustrated in Figure 3.1, which is then added to the respective model (surface mesh or probabilistic occupancy grid) to then produce an updated improvement map to iteratively and incrementally acquire measurements to estimate the models of the scene. This section of the thesis presents three such point selection processes. The first point selection process developed and proposed to actually drive the selective sensing strategy on a range sensor uses only the results of the surface mesh based improvement measure developed in Section 3.1. Subsection 3.3.1 develops this first selection process to demonstrate the additional improvement in overall accuracy of a scene representation that can be achieved by using the surface mesh based improvement measure as a basis in a

selective sensing process when compared to brute force scanning as well as random acquisition. A second selection process is proposed in Subsection 3.3.2, which is based on the probabilistic occupancy grid improvement measure, which as demonstrated in Section 3.2, provides a sense of coverage within the sensor field of view, therefore aiming at completeness rather than accuracy. Finally in Subsection 3.3.3, a third and final point selection process is presented that integrates both of the developed improvement measures in order to provide a compromise between accuracy and coverage in the automated data driven selective sensing process.

3.3.1 Surface Mesh Improvement Map Based Selection Processes

Assuming the operation of a sensor whose data is generated in a native coordinate system corresponding to the angular projection coordinates (defined in Section 2.4.1, and detailed in Appendix A), the improvement measure, $imp_{SM}(\theta_i, \varphi_i)$, developed in Section 3.1 provides an indication of how much estimated improvement there will be if the next point is acquired in a given sensing direction (θ_i, φ_i) assuming the current estimated depth is the actual depth, i.e. $\rho_i(\theta_i, \varphi_i) = \hat{\rho}_i(\theta_i, \varphi_i)$ in that direction. This first selection method, $DP_1(imp_{SM})$, determines the optimal target direction, (θ_i, φ_i), over all points, i, that belong to the set of unacquired points, P_U. That direction corresponds to the maximum value in the improvement map, as shown in eq. (3.83). If there exists multiple locations that have improvement levels equal to the maximum improvement value, only one of them is arbitrarily selected as the next location to measure. Applying this selection process to the example presented in Subsection 3.1.4 yields the next acquisition location to be $\hat{p}_1 = (0.03,\ 0.08)$, since it has the highest improvement value among the values calculated.

$$DP_1(imp_{SM}) = arg\max_{i \in P_U}\left(imp_{SM}(\theta_i, \varphi_i)\right) \tag{3.83}$$

This method can be extended to select multiple points, by ensuring that each selected point will not be in the region of influence of the other selected points, as otherwise the selected points would likely be direct neighbours, as well as violating the assumption made in the formulation of imp_{SM} of the improvement measure hypothesizing the addition of a single point in the local region. Due to the formulation of the algorithm, shown in Figure 3.5, these regions of influence correspond each to a separate triangle, T_k. Therefore the maximal improvement value location from each of the triangles that are contained in the surface mesh

can be selected simultaneously for acquisition in a multi-point variation of this algorithm. Furthermore, the acquisition of points can be constrained to only a maximum of N points, and/or to points that have corresponding improvement values greater than a fraction, f, of the maximum improvement value in the improvement map (see eq. (3.84)).

$$DP_1'(imp_{SM}, f, N) = \left\{ \hat{p}_1, \hat{p}_2, \dots, \hat{p}_k, \dots, \hat{p}_N \middle| \hat{p}_k = \underset{i \in P_U \cap T_k}{\arg\max} \left(imp_{SM}(\theta_i, \varphi_i) \in T_k \right), imp_{SM}(\theta_k, \varphi_k) \geq f \cdot imp_{SM}(\theta_1, \varphi_1), imp_{SM}(\theta_k, \varphi_k) \geq imp_{SM}(\theta_{k+1}, \varphi_{k+1}) \forall k \right\} \tag{3.84}$$

3.3.2 Probabilistic Grid Improvement Map Based Selection Processes

A selective sensing process is also developed using solely the improvement measure based on probabilistic occupancy grids, imp_{POG}, proposed in Section 3.2. This is accomplished by selecting the sensing direction, (θ_i, φ_i), which contains the maximum improvement value over the entire effective sensor range, S_R. Eq. (3.85) details this second selection process, $DP_2(imp_{POG})$.

$$DP_2(imp_{POG}) = arg \max_{\forall i \in S_R} \left(imp_{POG}(\theta_i, \varphi_i) \right) \tag{3.85}$$

Note that especially early in the acquisition process, there will not be a unique maximum, but there will instead be large regions containing locations which have improvement values equal to this maximum value. This point selection process will not select a unique location early in the single point of view acquisition process, but instead will choose an arbitrary point corresponding to this improvement value, and as such a guaranteed level of repeatability cannot be provided. Additionally, by using all points of a certain improvement level, instead of choosing an arbitrary one, it can be used in determining which regions have not yet been covered by the acquisition process, and additionally it can be used on nearly complete scenes to evaluate which regions need to be reacquired to increase certainty of occupancy.

A more general point selection process based on the same probabilistic improvement measure is developed by taking $DP_2(imp_{POG})$ presented in eq. (3.85), and altering it such that not only the maximum valued locations are selected, but also other high improvement regions as well. This new point selection process, $DP'_2(imp_{POG}, threshold)$, is accomplished by thresholding with an arbitrary improvement level, as presented in eq. (3.86). In fact, by

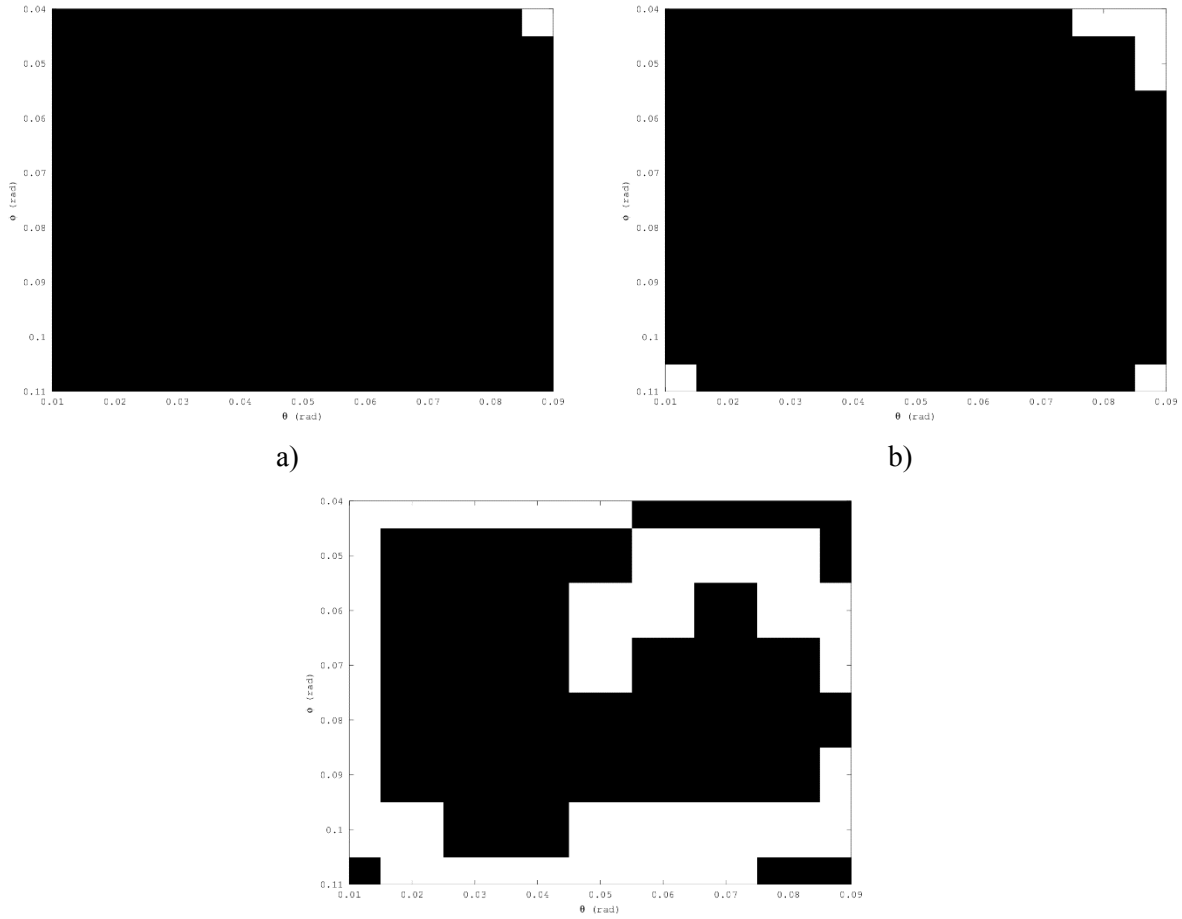
using this point selection process where the threshold is the maximum value, the same results as $DP_2(imp_{POG})$ are obtained.

$$DP'_2(imp_{POG}, threshold) = \{\hat{p}_1, \hat{p}_2, \dots, \hat{p}_k, \dots, \hat{p}_N | \hat{p}_k = (\theta_k, \varphi_k), imp_{POG}(\theta_k, \varphi_k) \geq threshold, \forall k \in S_R\} \quad (3.86)$$

Furthermore, by modifying this second selection process from selecting larger improvement valued regions, to selecting negative improvement valued regions that correspond to transition regions in the surface of highest probability of occupancy as stated in Section 3.2, a new selection process is created that uses negative thresholding instead of positive thresholding. This new selection process, $DP''_2(imp_{POG}, threshold)$, that will select regions of negative improvement is formulated in eq. (3.87).

$$DP''_2(imp_{POG}, \text{threshold}) = \{\hat{p}_1, \hat{p}_2, \dots, \hat{p}_k, \dots, \hat{p}_N | \hat{p}_k = (\theta_k, \varphi_k), imp_{POG}(\theta_k, \varphi_k) < threshold, \forall k \in S_R\} \quad (3.87)$$

If these selection processes were to be applied to the improvement map generated in Section 3.2.1, the regions (in white) in Figure 3.15 will be selected. Note that for the threshold cases (Figure 3.15 b) and c)), that multiple acquisition locations are selected, as expected.



a) b)

c)

Figure 3.15 - Results of executing the following selection processes: a) $DP_2(imp_{POG})$, b) $DP'_2(imp_{POG},4.7)$, and b) $DP''_2(imp_{POG},-3.5)$

3.3.3 Combined Improvement Measure Selection Processes

A combination of both improvement measures should see greater performance compared to the individual approaches in the two previous subsections. The surface mesh based approach when used alone has a drawback that it overly focuses on edges and transitions present in the scene, even to the extent of acquiring direct neighbours to already measured points. But, it produces an improvement map that has a gradient where there is usually only one location that has the maximum improvement value. On the other hand, the drawback to an approach that uses the probabilistic occupancy grid based improvement measure is that there are many locations that will be equal to the maximum improvement value leading to no explicit choice of a particular acquisition location, but it does produce higher improvement values further from currently acquired points. It therefore draws the attention towards less explored regions

of the scene, thereby favouring increased coverage. As the drawbacks of each improvement measure is balanced by the benefits of the other, by combining the resulting improvement maps together the selection process is able to select points which are not direct neighbours to currently acquired points, and simultaneously produces better coverage of interesting features throughout the scene.

The combination of both measures is accomplished by the element wise multiplication of the respective improvement maps. This ensures that locations close to already measured locations are only considered for acquisition if the surface mesh improvement dominates the effect of the low but non-negative improvement from the probabilistic occupancy grid approach. The negative improvement effect of the probabilistic occupancy grid approach, when present, will effectively push the selection of points to be further away from already acquired points. The negative improvement values in the probabilistic occupancy grid are removed when locations adjacent to the locations that contain negative improvement are measured, if they have similar depths. When they do not have similar depths (transitions, or a different feature) there will be constant negative improvement located on the transition. In this scenario, the drive for accurate representation is performed by selecting neighbouring locations that have a high surface mesh improvement value, which are directly adjacent to a negative probabilistic occupancy grid improvement value. Overall, this leads to the selection process to firstly drive acquisition in order to accurately represent the scene, while ensuring points are rarely selected to be adjacent to already measured points, and secondly ensuring that coverage of the scene is considered before final points are acquired to complete the accuracy driven representation of the scene. This point selection process, $DP_3(imp_{SM}, imp_{POG})$, is detailed in eq. (3.88).

$$DP_3(imp_{SM}, imp_{POG}) = arg \max_{i \in P_U} \left(imp_{SM}(\theta_i, \varphi_i) \cdot imp_{POG}(\theta_i, \varphi_i) \right) \tag{3.88}$$

There was another possible combined point selection process that could have been developed instead of the one presented in eq. (3.88) of this thesis. This other strategy that was briefly investigated was based on a linear combination of the two improvement measures. This was dismissed, as it is difficult to relate the two improvement metrics directly to each other in a generic manner, as the surface mesh improvement measure has different relative amplitudes depending on the scene, while the probabilistic occupancy grid improvement measure has

different relative amplitudes when using different sensor OPDF parameters. The weights in the linear combination scenario, would have been set in one of two ways, either a statically defined, user selectable value, which would have set the balance between the two methods, or a more automatic technique that would adjust the weights based upon the number of locations measured, such that earlier on in the acquisition process, accuracy via the surface mesh improvement map would dominate, and then later on coverage would dominate.

3.4 Chapter Summary

In Section 3.1, the development of the original proposed improvement measure was performed, starting with the formalism of Ordinary Kriging and the use of the VMR as way to express error. This resulted in eq. (3.51), which by itself is computationally intensive. This motivated the use of the approximation of the semivariogram discussed in Section 3.1.1, as it allowed the improvement to move from quadratic to linear performance with respect to the number of distance calculation needed for a local region with m unknown points. By including this approximation of the semivariogram, the final optimized surface mesh improvement measure was able to be developed in eq. (3.64). As the surface mesh improvement measure has larger improvement values about edges of object, transition regions, and other features, it drives acquisition, through the selection process developed in Section 3.3.1, towards the accurate modeling of the scene. The drawback to this method is that it tends towards acquiring locations adjacent to already measured locations when it occurs along an edge. Only when all the large transition edges have been found, does this selection process start in on improving coverage, as the larger improvement will then be located in the middle of the larger triangles that occupy the highly consistent regions.

As accuracy is not the sole desired criteria for driving a selective sensing process, a second original improvement measure was proposed based off of probabilistic occupancy grids, which was developed in Section 3.2. This improvement measure has larger improvement values located away from measure locations, promoting increasing coverage in the acquisition of the scene as driven by the selection process detailed in Section 3.3.2. The drawback to this method is that it tends to select regions as opposed to specific locations, although by taking into account the selection process, $DP''_2(imp_{POG}, \text{threshold})$, negative improvement can be used to drive accuracy to a limited degree.

Both the improvement measures, and their respective dedicated point selections processes, are highly oriented towards specific goals, namely accuracy for the surface mesh improvement measure, and coverage for the probabilistic occupancy grid measure. Two original point selection processes are proposed in Section 3.3.3 for combining both improvement methods in order to balance accuracy and coverage. One of them, $DP_3(imp_{SM}, imp_{POG})$, uses the product of the improvement maps at each location in order to determine where to acquire the next location. This provides the benefit of focussing on accuracy, but ensuring that the selected locations tend not to be directly adjacent to already measured locations, thereby encouraging coverage of the scene in the process.

The following chapters will discuss the experimental setup, and experimental validations of the proposed improvement metrics, as well as the selection processes, to demonstrate that they indeed perform as argued, while verifying that they will operate regardless of the sensor employed and scene captured.

Chapter 4. Experimental Setup

This chapter describes the experimental setup, in terms of the sensors employed, the parameter values for the improvement measures developed in Chapter 3 to be used for the experimental validation in Chapter 5, and the scenes captured. The first section of this chapter discusses the sensor configurations employed, in addition to determining the respective OPDF parameters for use in the probabilistic occupancy grid improvement map developed in Section 3.2. The second section details the value of the semivariogram approximation parameter, discussed in Section 3.1.1, used in this thesis. The third section provides details of the scenes used in the analysis of the algorithms developed in Chapter 3. Finally, the last section provides a qualitative analysis involving the sensors and the acquired data to highlight the significant issues in regards to the acquisitions.

4.1 Sensor Descriptions

This subsection describes six different sensor configurations that will be used for data acquisition within this thesis. The primary reason for using several different sensors is to affirm that the techniques developed in Chapter 3 are indeed independent of the sensor used for acquisition. The six sensors are the Microsoft Kinect, an in-house adaptive structured light stereo vision system (ASLS), an in-house integrated robotic laser range finder system, a Jupiter laser range scanner mounted on a track, a SICK LMS mounted on a track, and the Neptec Laser Metrology System (LMS). While the Jupiter and SICK LMS are being moved by the robot arm or motorized track in order to acquire a full scene, these integrated sensor systems can be considered as if they were a single sensor. This is due to the fact that the robot arm or the motorized track are only modifying one degree of freedom of the sensor's pose, namely rotation about the center acquisition axis when mounted on the robotic arm, and translating along the axis perpendicular to the line of acquisition when mounted on the motorized track, hence resulting in the data being able to be represented as if it were obtained from a single point of view. The six sensor configurations are described in detail, and their corresponding error models are used to determine the OPDF parameters that are employed in this thesis. This information is then summarized in Section 4.1.7 with Table 4.1.

4.1.1 Microsoft Kinect

The Microsoft Kinect, as seen in Figure 4.1, is a multi-modal sensor [82], which is capable of providing depth images through an infra-red (IR) structured light system, an IR image, and a traditional colour (RGB) image, plus it contains a microphone array as well as a tilt mechanism. The Kinect is developed using the technology of PrimeSense [83,84], and was designed for the purpose of interactive gaming on the Microsoft XBox 360 console. But it has attracted the interest of robotics researchers and hobbyists as an inexpensive sensor platform by using the OpenNI open source Kinect drivers [85], as well as third parties manufacturing similar sensors. The data of interest to this thesis is the depth image, which can be set to 320x240 or 640x480, with a 57° horizontal, and a 43° vertical field of view. Since the Kinect is a stereo system, which acquires data using a projective geometry, the angular projection coordinate system will be used. The reliable range of operation is between 0.8m and 2.5m [82,84,86], but it is capable of acquiring data within the larger range of 0.5m to 9.0m. For the purposes of this thesis, the range considered is 0.8m – 5.0m in order to ensure sufficient scene coverage, while minimizing error. The Kinect produces a dense dataset quickly, but it has fairly large errors compared to other sensors, and it is limited in its usage for scenes close to the sensor and without a large range of depths. The Kinect performs poorly in scenes with abundant natural light, and dark surfaces [87].

Figure 4.1 - Microsoft Kinect

According to eq. (4.1) developed by Khoshelham [86], the error in the depth sensor, σ_Z, is primarily along the depth axis, and is proportional to the product of square of the distance, ${Z_k}^2$, and the disparity quantization error, σ_d. The proportionality constant is a function of the camera focal length, f, the baseline between the IR projector and the IR camera, b, and

an additional calibration parameter, m, used for scaling disparity to distance. Khoshelham's error model for the Kinect produces a variance along the depth axis of 2.23×10^3 mm^2 at distance of 5.0m. Additionally, due to the triangulation nature of the depth measurements, any error along the horizontal and vertical axis will result in a depth error, and hence error along these two axes can be modelled by quantization noise. The specifications for the Kinect detail that the spatial resolution at 2m depth is 3mm along the horizontal and vertical axis. The angular width and height of pixels in perspective images are larger at the centre versus the edges. In this thesis, using a single static variance for modelling the OPDF is desired to minimize system complexity, and as such only the maximum variances of the sensors parameterization are considered. Due to these considerations, the maximum variance, which corresponds to the largest angular width and height of a pixel, occurs at the centre of the field of view of the Kinect. This horizontal and vertical variance is calculated by assuming that the quantization error is uniformly distributed. These angular variances are calculated by first solving the trigonometric system illustrated in Figure 4.2, which results eq. (4.2) where α is determined to be 1.5×10^{-3} rad. This value for α is then used in eq. (4.3), which is the formulation for determining the variance of the uniform distribution [88]. The final horizontal and vertical angular variances are found to be 1.875×10^{-7} rad^2.

$$\sigma_Z = \left(\frac{m}{fb}\right) {Z_k}^2 \sigma_d \tag{4.1}$$

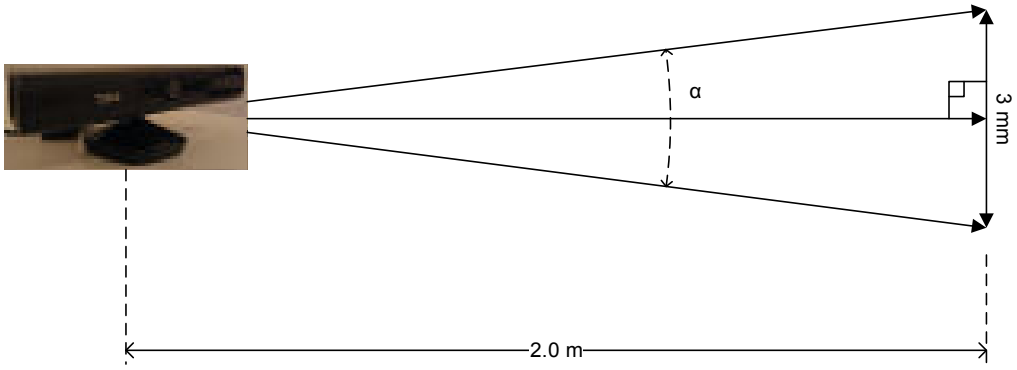


Figure 4.2 - Kinect horizontal and vertical error

$$\alpha = 2 \cdot atan\left(\frac{3/2}{2000}\right) = 1.5 \times 10^{-3} \tag{4.2}$$

$$\sigma_\theta{}^2 = \sigma_\varphi{}^2 = \frac{\alpha^2}{12} = 1.875 \times 10^{-7} \tag{4.3}$$

4.1.2 Adaptive Structured Light Sensor

Boyer [89,90] develops an adaptive structured light sensor (ASLS), shown in Figure 4.3, based upon the work of Desjardins [91], which involves projecting a marching pseudo-random pattern on to a scene to act as a generator for features for a set of stereo cameras. Desjardins uses three colours to produce a pattern, where any 3x3 code block is unique. The center of the code is used as the feature point for the stereo matching. Furthermore, a Hamming distance for error correction is included which allows for the capturing of small features and edges, which a whole pattern could not cover entirely. Boyer extends and improves the work by introducing time domain multiplexing of the pattern, such that three white sub patterns are used, which each pattern corresponding to one of the three colours, and projected at three different instances in time, which improves the detection of the pattern on a wider variety of surfaces. In addition, Boyer adds multi-focal capability, and increases the maximum possible resolution of the acquisition by shifting the pattern over several iterations horizontally and vertically. The sensor provides dense and precise depth readings over a wide variety of surfaces, but it takes a long time to acquire a scene in high detail.

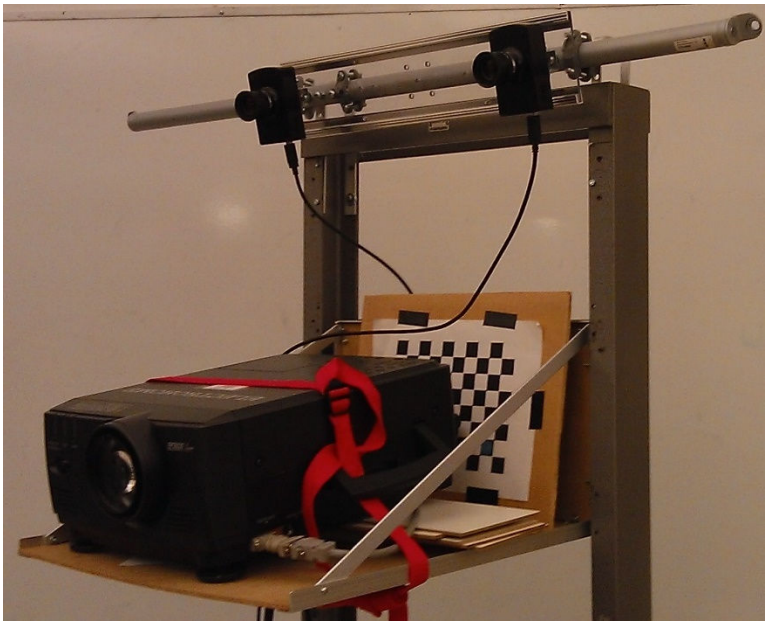


Figure 4.3 - Adaptive structured light sensor

As the ASLS uses projective geometry, the angular projection coordinate system will be used to represent the data. The error model is similar to the Kinect's, as they are both

triangulation based sensors, but due to the higher resolution cameras employed, and the sub-pixel resolution that the projected codes allow, the quantization error is much lower than the Kinect's. Due to the bulkiness of the sensor, the fairly large baseline, and the positioning of the projector to ensure sufficient coverage of the scene, the range considered for the ASLS, for the purposes of this thesis, is 1.0m – 10.0m. The ASLS system for the acquisition of data for this thesis is configured with lenses that have focal lengths, f_x and f_y, of approximately 1830 measured in pixels, cameras which acquire images with a resolution of 1392x1040, and a baseline between the pair of cameras, b, of 400 mm as determined by the calibration routine that is powered by OpenCV [92]. This yields a maximal field of view of approximately 41.4° horizontally and 31.7° vertically. The smallest feature detected and admissible to be considered part of the 3x3 code by the ASLS is 5 pixels. Assuming that the feature is approximately square, this implies that in the worst case, the feature is approximately 2x2 pixels, which means that the center, which is used as the location of the feature, will have a precision of approximately half a pixel. This implies that the disparity value, d, will have an error of approximately 0.5 pixel, which can be assumed to be uniformly distributed. This implies that according to the equation for the variance of the uniform distribution, ${\sigma_d}^2$ is 0.0208 pixel^2. Using the relationship between depth and disparity, shown in eq. (4.4), the maximum depth, 10m, and the disparity error standard deviation, the variance for the depth can be calculated as the square of the difference between 10m and the distance corresponding to the disparity at 10m plus twice the disparity standard deviation error, as shown in eq. (4.5). This results in the theoretical worst case depth quantization error of 39.3 mm at 10 m distance, and a depth variance, ${\sigma_\rho}^2$, of 129 mm^2.

$$z = \frac{bf_x}{d} \tag{4.4}$$

$${\sigma_\rho}^2 = {\sigma_z}^2 = \left(\frac{2z^2\sigma_d}{fb + 2z\sigma_d}\right)^2 \tag{4.5}$$

As previously stated for the Kinect in Section 4.1.1, the angular width and height of pixels in perspective images are larger at the centre versus the edges. Since only a single static variance for modelling the OPDF is desired, therefore only the maximum variances of the sensors parameterization are considered, which corresponds to the largest angular width and height of a pixel that occurs at the centre of the field of view of the reference camera in the

ASLS. This horizontal and vertical variance is calculated by assuming that the quantization error is uniformly distributed. The horizontal and vertical angular variances are calculated by solving the trigonometric system illustrated in Figure 4.4, which results in eq. (4.6) where α is determined to be 2.71x10-4 rad. This results in the angular variances, as determined by eq. (4.3), $\sigma_\theta{}^2$ and $\sigma_\varphi{}^2$, equaling 6.14x10^{-9} rad^2.

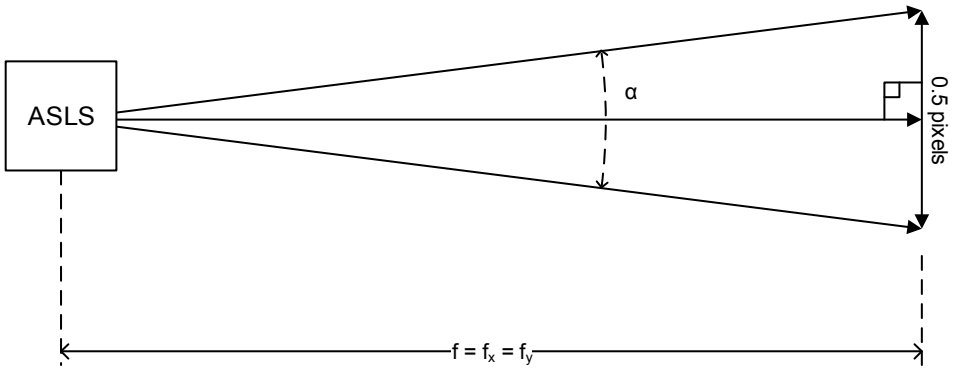


Figure 4.4 - ASLS horizontal and vertical errors

$$\alpha = 2 \cdot atan\left(\frac{0.5/2}{f}\right) = 2.71 \times 10^{-4} \tag{4.6}$$

4.1.3 Jupiter Laser Range Finder Mounted on F3T Robot Arm

The integrated robotic laser range sensing system developed by Curtis [93,94] and shown in Figure 4.5, integrates a F3T manipulator from CRS Robotics with a Jupiter laser range finder manufactured by Servo-Robot based on active triangulation technology developed at NRC [95]. The Jupiter laser range finder is capable of acquiring 256 or 512 points along a single line with a field of view encompassing 1005 mm field depth, a 108 mm wide near plane, and a 579 mm wide far plane, at a depth resolution of 0.7 mm [96]. Data is sampled angularly uniform by the sensor. The integration of this sensor in place of the end effector on the F3T robotic arm, allows for the capture of a scene by either rotating the sensor, tilting it, or by moving the robotic arm along a particular path. For the purposes of data collection within this thesis, the Jupiter laser range finder is rotated solely about the wrist of the F3T robotic arm (no translation) during acquisition, which yields a spherical native geometry. The integrated system provides high precision measurements with a variable density controlled by the robotic motion, but it does not have a wide field of view compared to other sensors,

and tends to produce outliers on surfaces that do not reflect well the laser beam, in addition to edges.

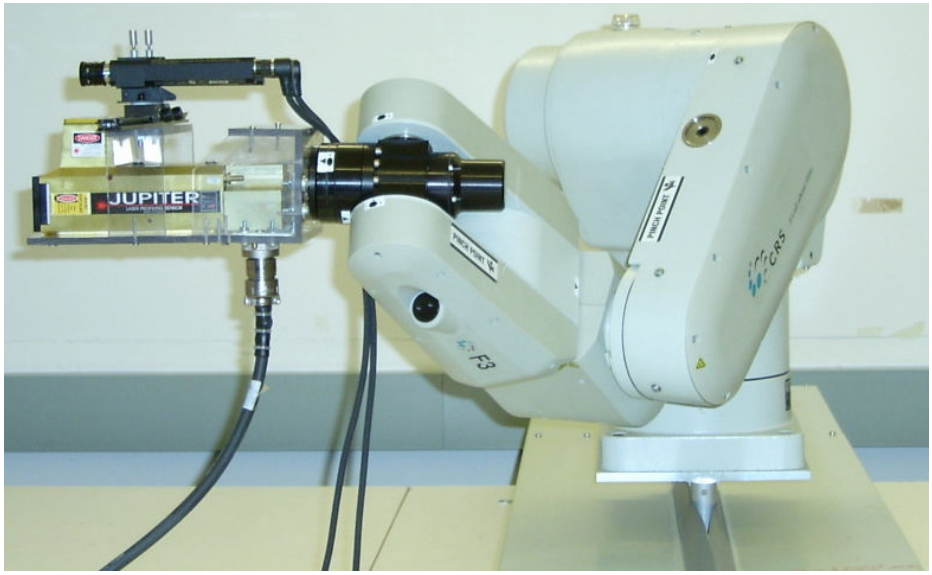


Figure 4.5 - Integrated robotic laser range sensing system

Using the assumption that the quantization error along the depth axis is uniformly distributed, then the depth variance, $\sigma_\rho{}^2$, equals 4.08x10^{-2} rad^2 as performed in eq. (4.7). Along the elevation axis, the standard deviation has been determined to be 2x10^{-3} rad, and hence results in a variance, $\sigma_\varphi{}^2$, of 4x10^{-6} rad^2. The azimuth error is determined by the angular error of the F3T's end effector, which consists of ± 5 pulses, with a gearing ratio of 80:1, and 2048 pulses per rotation [97]. Assuming that the error is uniformly distributed, then the azimuth variance, $\sigma_\theta{}^2$, is calculated as shown in eq. (4.8), which equals 1.23x10^{-8} rad^2.

$$\sigma_\rho{}^2 = \frac{0.7^2}{12} = 4.08 \times 10^{-2} \tag{4.7}$$

$$\sigma_\theta{}^2 = \frac{\left((2 \times 5) \cdot \frac{2\pi}{(2048 \cdot 80)}\right)^2}{12} = 1.23 \times 10^{-8} \tag{4.8}$$

4.1.4 Jupiter Laser Range Finder Mounted on Linear Motorized Track

The Jupiter laser scanner has also been mounted on a linear motorized track (see Figure 4.6). In this setup, the native sensor geometry is cylindrical. Since the Jupiter sensor is the same

as in Section 4.1.3, the depth variance is the 4.08×10^{-2} mm^2 and the angular variance is 4×10^{-6} rad^2. For the purposes of this thesis, the track was incremented in 1mm steps, for 550 steps, and the track accuracy with this step size is ±0.1 mm. Assuming that this ±0.1 mm position error is uniformly distributed, then the variance, $\sigma_x{}^2$, is calculated to be 3.33×10^{-3} mm^2 along the translational axis, as show in eq. (4.9). Again, as previously stated in Section 4.1.3, this sensor has high precision, but it does not have a wide field of view compared to other sensors, and tends to produce outliers on surfaces that do not reflect well the laser beam, in addition to edges. The addition on the linear motorized track allows for more of a scene to be covered, but as the track only allows for a 550mm total displacement, the overall sensor is limited to relatively small scenes.

$$\sigma_x{}^2 = \frac{(2 \times 0.1)^2}{12} = 3.33 \times 10^{-3} \tag{4.9}$$

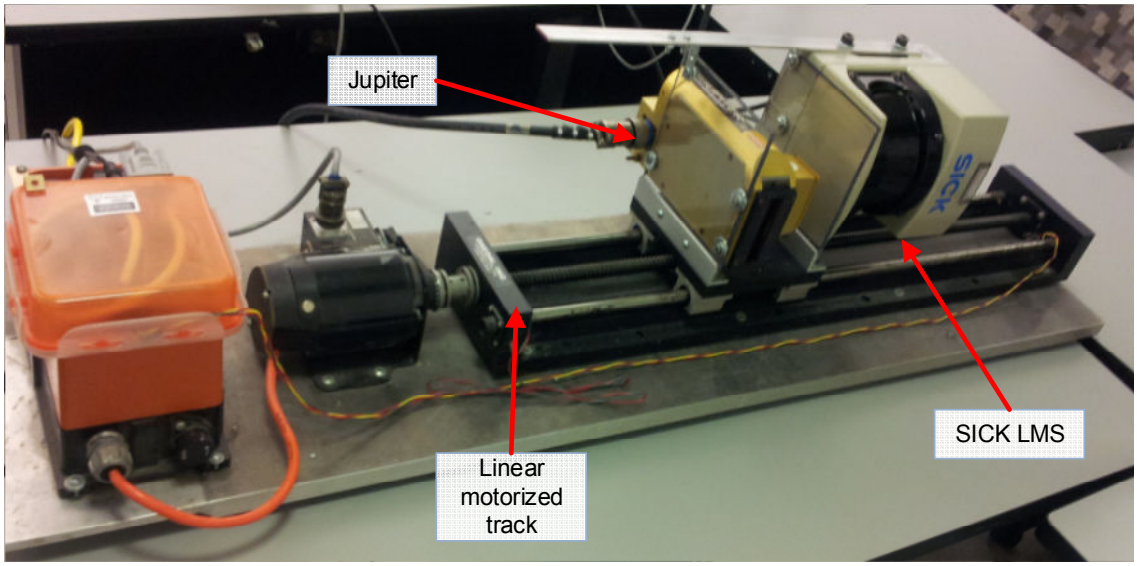


Figure 4.6 - SICK LMS and Jupiter in tandem on track

4.1.5 SICK LMS Mounted on Linear Motorized Track

The SICK LMS 291-s14, as seen in Figure 4.7, is an infrared laser line scanner which has a 90° field of view, and an angular resolution of 0.5°, which means that it is capable of sampling 181 angularly uniform points within its field of view [98]. The operational range is between 0 and 80 m, with a maximal error on depth of ± 35mm. The SICK LMS can be used in outdoor environments, and is quite versatile in regards of surface types that can be acquired, although it underestimates the depth at edges. Since it can only acquire data along

a single line, just as in the case of the Jupiter, it must be moved to acquire a 3D scene. The same linear motorized track that is used in Section 4.1.4, is also used for this SICK LMS, with it being mounted in tandem with the Jupiter sensor. While the SICK LMS has a much wider field of view compared to the Jupiter, it has less angular resolution.

Since the same track is used for the SICK LMS as was used for the Jupiter, the translational axis variance is the identical, at 3.33×10^{-3} mm^2, with the same operational parameters of 550 steps in 1mm increments. For this thesis, only the range 1mm-8000mm is considered. The error within this range is dominated by the quantization error, which is 10.0 mm. Assuming that the quantization error is uniformly distributed, this yields a depth variance, $\sigma_\rho{}^2$, of 8.33 mm^2, as shown in in eq. (4.10). The angular resolution of the sensor is 0.5°, and assuming that the quantization error of the angular displacement is uniform, this results in the angular variance, $\sigma_\theta{}^2$, being equal to 6.35×10^{-6} rad^2, as shown in eq. (4.11).

Figure 4.7 - SICK LMS 291-s14 range finder

$$\sigma_\rho{}^2 = \frac{(10)^2}{12} = 8.33 \tag{4.10}$$

$$\sigma_\theta{}^2 = \frac{\left(0.5\frac{\pi}{180}\right)^2}{12} = 6.35 \times 10^{-6} \tag{4.11}$$

4.1.6 Neptec Laser Metrology Sensor

The Neptec Laser Metrology Sensor (LMS), shown in Figure 4.8, is a 3D asynchronous laser range sensor capable of guiding acquisition in preset patterns as well as being capable of random access range sensing. The technology behind this sensor was developed through the

NRC, and is based on the Laser Camera System developed by Neptec Design Group Ltd. for the NASA space shuttle program [99]. The field of view is 30° by 50° and has a depth error of 1.2mm at 5m, and a horizontal and vertical error of 73 μm. As such, this sensor is capable of precise and dense customizable 3D acquisitions. For this thesis, only a single dataset was acquired with this sensor. This scene was acquired with a 22.5°x22.5° field of view, at a resolution of 1024x1024, and only a depth range of 600mm-5000mm is considered. The native sensor geometry can be considered to be projection based. The error in depth is assumed to be uniformly distributed, yielding a variance, $\sigma_\rho{}^2$, of 0.12 mm^2, as shown in eq. (4.12). Furthermore, the angular variances are calculated by solving the trigonometric system presented in Figure 4.9, for α, which equals 1.46x10^{-5} rad as shown in eq. (4.13). By assuming that these angular errors are uniformly distributed, the variances, $\sigma_\theta{}^2$ and $\sigma_\varphi{}^2$, are calculated to be 1.78x10^{-11} rad^2, as shown in eq. (4.14). The Neptec LMS produces precise measurements that have few outliers.

Figure 4.8 - Neptec LMS

$$\sigma_\rho{}^2 = \frac{(1.2)^2}{12} = 0.12 \tag{4.12}$$

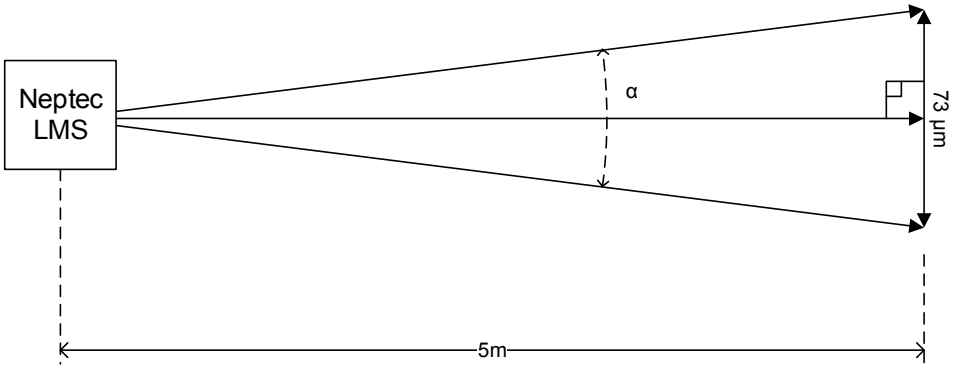


Figure 4.9 - Neptec LMS horizontal and vertical errors

$$\alpha = 2 \cdot atan\left(\frac{73/2}{5 \times 10^6}\right) = 1.46 \times 10^{-5} \tag{4.13}$$

$$\sigma_\theta{}^2 = \sigma_\varphi{}^2 = \frac{(\alpha)^2}{12} = 1.78 \times 10^{-11} \tag{4.14}$$

4.1.7 Summary of Sensor Parameterizations

Table 4.1 summarises the OPDF parameters for the sensor configurations listed in the Sections 4.1.1 - 4.1.6. The first column of the table details the parameter being listed, where *Native Sensor Geometry* is the native geometry of the sensor configuration being employed in this thesis, C_0 *Resolution* is the discrete resolution corresponding to the first axis of the native sensor geometry, as presented in Section 2.4.1, and further detailed in Appendix A. Similarly, C_1 *Resolution* and C_2 *Resolution* are the discrete resolutions corresponding to the second and third axis of the native system geometry. Furthermore, the C_0 *Range*, C_1 *Range*, and C_2 *Range* correspond to the field of view of the sensor in the native coordinate system, while C_0 *Variance*, C_1 *Variance*, and C_2 *Variance* correspond to the variances of the sensor configurations along their native coordinate systems.

	Kinect	ASLS	Jupiter/F3T	Jupiter/Track	SICK/Track	Neptec
Native Sensor Geometry	Projective	Projective	Spherical	Cylindrical	Cylindrical	Projective
C_0 Resolution	640	1392	180	550	550	1024
C_1 Resolution	480	1040	512	512	181	1024
C_2 Resolution	512	512	512	512	1500	512
C_0 Range	57°	41.4°	180°	-550-0mm	-550-0mm	22.5°
C_1 Range	43°	31.7°	35°	35°	90°	22.5°
C_2 Range	0.8-5.0m	1-10m	170-1175mm	170-1175mm	1-8000mm	600-5000mm
C_0 Variance	$1.88\text{x}10^{-7}$ rad^2	$6.14\text{x}10^{-9}$ rad^2	$1.23\text{x}10^{-8}$ rad^2	$3.33\text{x}10^{-3}$ mm^2	$3.33\text{x}10^{-3}$ mm^2	$1.78\text{x}10^{-11}$ rad^2
C_1 Variance	$1.88\text{x}10^{-7}$ rad^2	$6.14\text{x}10^{-9}$ rad^2	$4\text{x}10^{-6}$ rad^2	$4\text{x}10^{-6}$ rad^2	$6.35\text{x}10^{-6}$ rad^2	$1.78\text{x}10^{-11}$ rad^2
C_2 Variance	$2.23\text{x}10^{3}$ mm^2	129 mm^2	$4.08\text{x}10^{-2}$ mm^2	$4.08\text{x}10^{-2}$ mm^2	8.33 mm^2	0.12 mm^2

Table 4.1 - Sensors OPDF parameters

4.2 Semivariogram Approximation Parameter

The previous section mentions the sensors, and their derived OPDF parameters that are being used in this thesis. The surface mesh improvement measure, developed in Section 3.1, uses the same range and resolution parameters as the probabilistic occupancy grid improvement measure listed in Table 4.1, but there is one additional parameter that needs to be defined. This additional parameter is the semivariogram approximation parameter, ϵ. This thesis sets the value of this parameter to be 0.00001, which means that the exponent of the power model defined in Section 2.2.7 is $w = 1.99999$. This value was chosen during testing as it provides a low approximation error while being large enough to avoid most instances of ill-conditioned K matrices for Kriging. Figure 4.10 shows the magnitude error, and Figure 4.11 shows the percentage error between the semivariogram models h^2 and $h^{2-\varepsilon}$, which contains the one of the cases presented in Figure 3.2 and Figure 3.3, where $\epsilon = 0.001$, the case where $\epsilon = 0.0001$, and the case being used in the evaluation, $\epsilon = 0.00001$. Notice that both errors

are pretty much negligible for $\epsilon = 0.00001$, with the percentage error being less than 0.02% and the magnitude error being less than 0.0004 in the range studied.

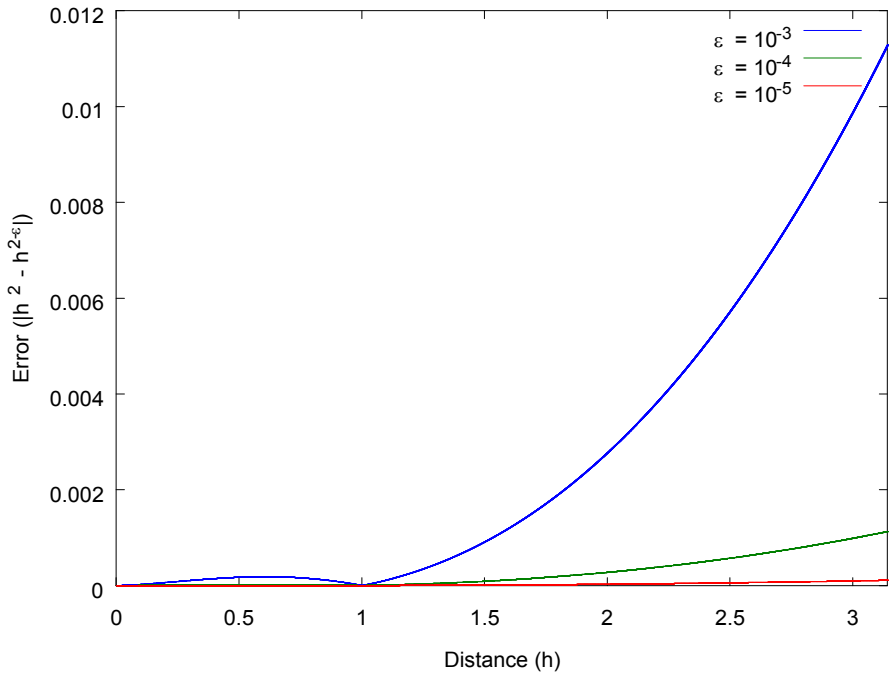


Figure 4.10 - Magnitude error between h^2 and $h^{2-\varepsilon}$ for ε=10^{-3}, 10^{-4}, and 10^{-5}

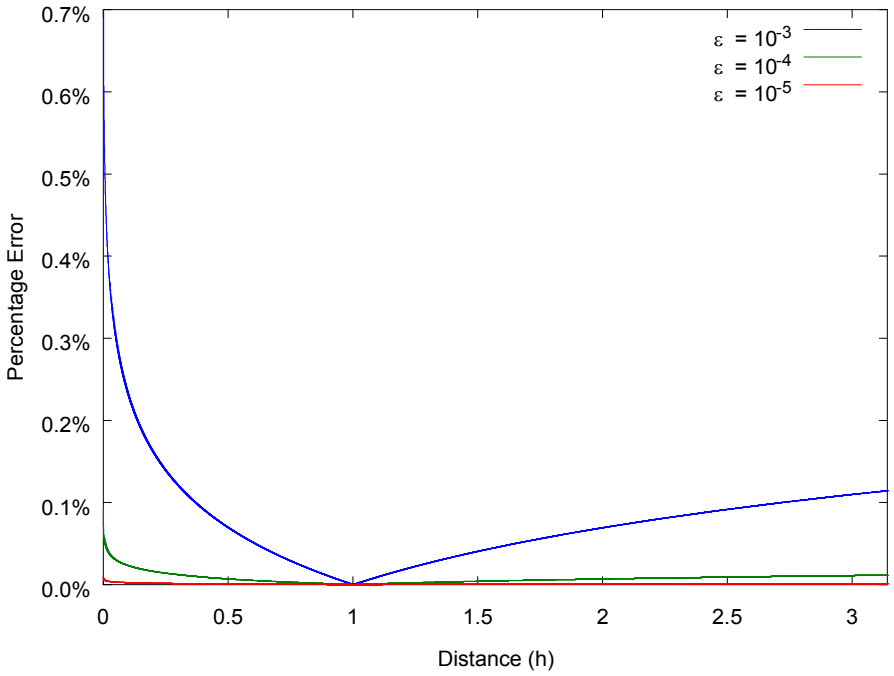


Figure 4.11 - Percentage error between h^2 and $h^{2-\varepsilon}$ for ε=10^{-3}, 10^{-4}, and 10^{-5}

4.3 Range Image Datasets

Several scenes were acquired by the different range sensors, which are described in Section 4.1, for the purpose of evaluating the algorithms developed in Chapter 3. Not all sensors acquired data from all scenes, due to several reasons including suitability of the scene for that particular sensor, surface properties of the objects that inhibited acquisition, and unnoticed faulty collection at acquisition time (human error). The 22 scenes that were acquired are fully presented and detailed in Appendix C. Table 4.2 provides the mapping of which sensor acquired each particular scene, in addition to information on the sections in Chapter 5 where that particular scene and sensor combination are used in the experimental evaluation. The scenes were chosen to ensure that as many different surface types and feature densities were presented to the sensors as possible. This includes scenes containing convex and concave curvatures, planar surfaces, objects with sparse acquisition due to inability of the sensor to acquire a surface, discontinuities, small scale features, edges, interior corners, and exterior corners.

Scene	Kinect	ASLS	Jupiter/F3T	Jupiter/Track	SICK/Track	Neptec
1	*i*					
2	*i*					
3	*i*					
4	*i*					
5	*i*					
6	*i*					
7	*ii, iii*	*ii, iii*	*ii*		*ii*	
8	*ii*	*ii*	*ii*	*ii*	*ii*	
9	*ii*	*ii*	*ii*		*ii*	
10	*ii*	*ii*	*ii*	*ii*	*ii*	
11	*ii, iii*	*ii, iii*	*ii*	*ii*	*ii*	
12	*ii*	*ii*			*ii*	
13	*ii*	*ii*	*ii*	*ii*	*ii*	
14	*ii*	*ii*				
15	*ii*		*ii*		*ii*	
16	*ii*	*ii*	*ii*		*ii*	
17	*ii*	*ii*	*ii*		*ii*	
18	*ii*	*ii*	*ii*	*ii*	*ii*	
19	*ii*	*ii*	*ii*		*ii*	
20	*ii, iii*	*ii, iii*		*ii*	*ii*	
21	*ii*	*ii*			*ii*	
22						*ii*

Notes:

i) *used for the evaluation in Sections 5.2 and 5.3*

ii) *used for the evaluation in Section 5.4*

iii) *used for the comparison of the surface mesh improvement measure and the neural gas selection method in Section 5.2.4*

Table 4.2 - Map of which sensors acquired the specified scenes, and the sections where the specific datasets are used

4.4 Experimental Setup Qualitative Analysis

This section performs a qualitative analysis of each sensor, by providing a raw 3D point cloud acquired by each sensor, and the native representation depth map. The 3D point cloud is rendered, manipulated, and captured using MeshLab [100]. The depth map, on the other hand, is produced by setting the appropriate values for the depth for each point location in the native geometry of the sensor configuration. Black pixels correspond to no acquisitions made at those particular locations, lighter gray pixels correspond to deeper values, and darker gray pixels correspond to shallower depth values. The drawbacks of each sensor are highlighted analysing these representations.

As discussed in Section 4.1.1, the Kinect does not produce many outliers, and most of the errors present in the resulting data is attributed to quantization error. Figure 4.12 shows the raw 3D data of the computer scene plotted in Cartesian space in the left image, and the native representation depth map in the right. The bumpiness of the background wall in the 3D representation is a result of the quantization error of the Kinect, and the black regions in the depth map correspond to sections where the Kinect cannot acquire data, either due to the surface reflectivity (as can be seen in the middle of the monitor), or due to the fact that the IR camera of the Kinect does not have line of sight over all surfaces on which the structured light pattern is projected (as illustrated by the left side of the computer case and monitor).

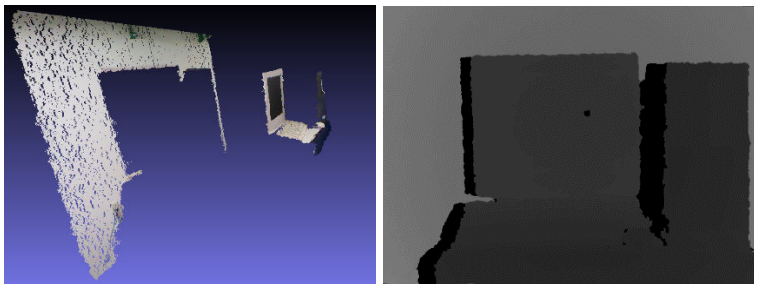


Figure 4.12 - Kinect acquisition of computer scene: 3D data (left) and native representation depth map (right)

The ASLS, which was presented in Section 4.1.2, tends to produce many outliers due to incorrect pattern matching that occurs due to refractions, reflections, and misalignment of the projected light pattern. The error due to quantization error is present but not as severe as the outliers, as can be seen in Figure 4.13, which shows the raw 3D data from the computer scene plotted in Cartesian space in the left image, and the native representation depth map in the right image. An example of the reflections/refractions of the projected pattern producing outliers occurs on the monitor surface, which produces larger shifts in the disparity between codes than what actually occurs. Additionally, the lack of precise representation of edges and narrow surfaces, such as the edge of the computer monitor, is due to the inherent nature of using a large 3x3 pattern as the matching feature that means that there is less surface area on which features are able to be projected. The reason for the large black regions around the computer in the depth map, is due to the projector having a narrower field of view than the cameras themselves thereby only a subset of locations within the field of view are able to be measured. Unfortunately, due to the fact that the projector may be repositioned anywhere

within the cameras' full field of view on a scene-to-scene basis, the full resolution range of the cameras' must be considered for this thesis.

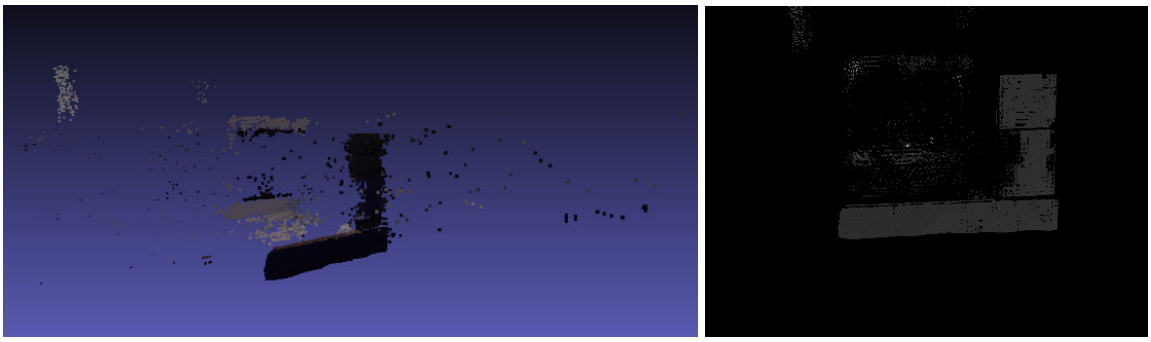


Figure 4.13 - ASLS acquisition of computer scene: 3D data (left) and native representation depth map (right)

The integrated robotic laser range sensing system that was presented in Section 4.1.3 exhibited the fact that the Jupiter sensor has high precision, but does not have a wide field of view compared to the other sensors. Figure 4.14, which shows the raw 3D data of the computer scene plotted in Cartesian space in the left image, and the native representation spherical depth map in the right image. As can be observed in the 3D image on the left, the glass on the monitor is not easily captured beyond a certain angle, and the Jupiter produces outliers in that vicinity. Notice that the field of view is a narrow cone, and this is a direct result of the Jupiter being rotated about the axis of the F3T robot arm, but as can be seen in the spherical depth map on the right, the regions that are visible to the sensor are captured. While the spherical mapping is more difficult for a human to identify features, it is still possible to identify them as long as the realization that the representation can be thought of as though each scan line is being represented as a full column of the depth image, with each separate column representing a different angle of rotation. The keyboard is captured in the top left and bottom right, the centre of the monitor is the large gray feature in the middle, and the sinusoidal thick lines across the bottom and top are the edges of the monitor.

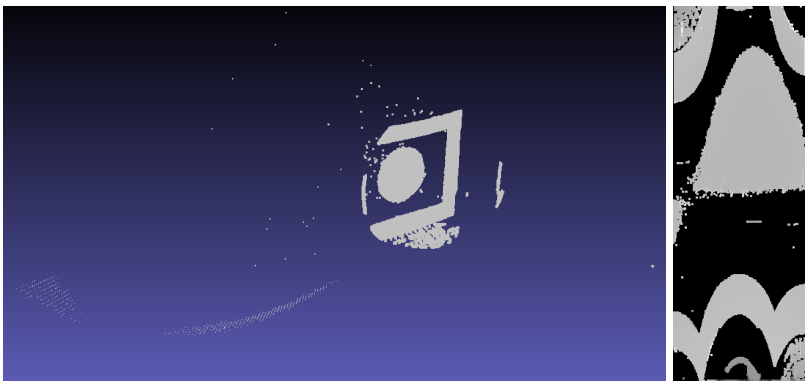


Figure 4.14 - Jupiter mounted on F3T acquisition of computer scene: 3D data (left) and native representation depth map (right)

The Jupiter sensor mounted on the linear motorized track, as presented in Section 4.1.4, has high precision but produces outlier at edges and on surfaces that are highly reflective and absorptive to the laser. Figure 4.15 shows the raw 3D data of the computer scene plotted in Cartesian space in the left image, and the native representation depth map in the right image. The 3D representation, shown on the left, highlights the many such outliers captured due to the surface being acquired, as well as the edges of the surfaces. Compared to the acquisition made by the previous configuration of the Jupiter mounted on the F3T robotic arm in Figure 4.14, the acquisition made by the Jupiter mounted on a track yields a larger field of view.

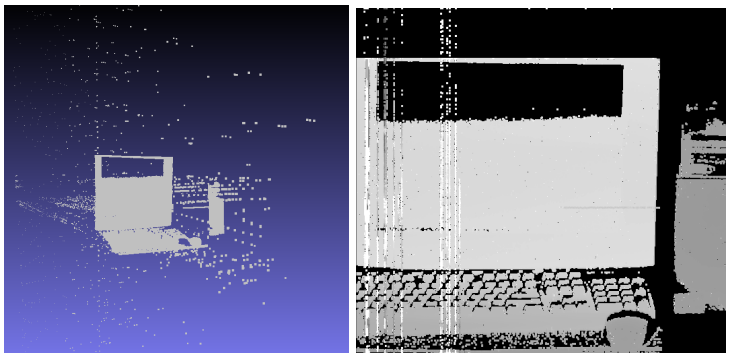


Figure 4.15 - Jupiter mounted on track acquisition of computer scene: 3D data (left) and native representation depth map (right)

The SICK LMS mounted on the linear motorized track that was detailed in Section 4.1.5 is able to acquire data with a large field of view and fairly high precision. Figure 4.16 shows the raw 3D data of the computer scene plotted in Cartesian space in the left image, and the native representation depth map in the right image. The vertical axis of the native representation appears highly compressed, as there is a much larger field of view with a

lower sampling rate than that for the horizontal axis. The region enlarged in the left image illustrates the drawback to the SICK LMS, which is to generate closer depth values than surrounding region at edges of objects, but fortunately the sensor produces few outliers, and is able to acquire data on surfaces that the other sensors had difficulty with, such as the glass on the computer monitor.

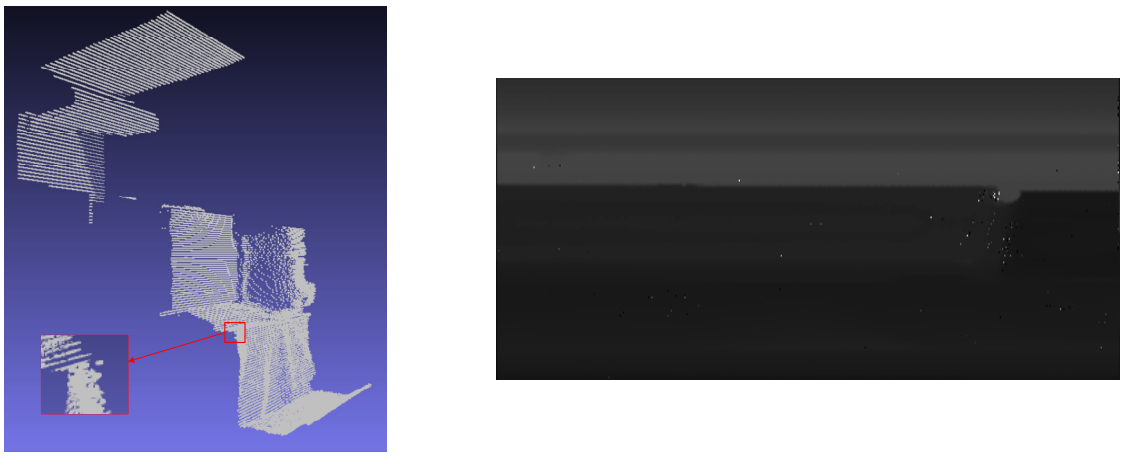


Figure 4.16 - SICK LMS mounted on track acquisition of computer scene: 3D data (left) and native representation depth map (right)

Finally, the Neptec LMS, which was presented in Section 4.1.6, produces few outliers, and is fairly precise. This can be observed in Figure 4.17, which shows the raw 3D data of a car door panel segment plotted in Cartesian space in the left image, and the native representation depth map in the right image. The door handle and seem are picked up well, as can be seen in the 3D representation (the depth variation is not well shown in the depth map due to the full depth range being compressed for display purposes, although the handle can be made out) .

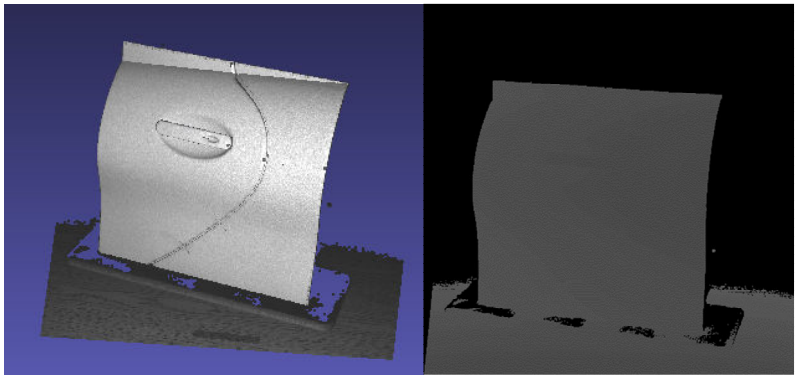


Figure 4.17 - Neptec LMS acquisition of a door panel segment: 3D data (left) and native representation depth map (right)

Chapter 5. Experimental Evaluation and Analysis

This chapter provides the results of applying the developed algorithms from Chapter 3 to range data acquired from a variety of sensors presented in Chapter 4 and detailed in Appendix C. The first section describes the experimental procedures and metrics used for the evaluation of proposed algorithms. The second section details the experimental evaluation of the surface mesh improvement based selection process while the third and fourth sections present the evaluation of the probabilistic occupancy grid improvement based and combined improvement measure selection processes respectively. The final section discusses the overall evaluation of the techniques.

5.1 Experimental Procedures and Metrics

This section details the experimental evaluation procedures for the techniques developed in Chapter 3. Additionally, the metrics that are used in the evaluations in the remaining sections of Chapter 5 are presented, with descriptions of what they are, how they are obtained, and the expected behaviours. A uniformly distributed random point selection process is chosen as the baseline comparison technique, as it produces points that are uniformly distributed within the scene during the acquisition process, but it requires no underlying data feedback mechanism. In other words, random point selection corresponds to a naïve selective sensing technique, which is essential for comparing a data driven technique against.

The procedure for the evaluation, unless explicitly stated otherwise for a particular evaluation, is as follows:

1) If using any of the selection processes except for random point selection, the raw 3D data points are binned in the sensor's native coordinate space, according to the field of view and resolution of the sensor. If there are more than one point in the bin, the point closest to the center of the bin is chosen to represent the bin location, and if there are no points in a bin, the corresponding location is considered as an unacquirable location (imperceptible to the sensor). For random point selection, the order that all of the raw 3D acquired points are chosen to be integrated into the respective improvement algorithms at random. This process can yield a small discrepancy between the number of points chosen by random selection and the other

point selection processes. Furthermore, this arrangement of the real raw data is treated as the individual locations that the sensor can acquire, and is treated as if it is capable of random access, for the purposes of this thesis.

2) An initial 10x10 uniformly distributed subsampling of the range data within the sensor's native coordinate space is performed.
3) These acquired points are then integrated into both the surface mesh and probabilistic occupancy grid models, and their respective improvement maps are determined. The calculations for each model are performed in parallel, and the durations of their respective executions are recorded. Locations in the acquisition that are selected, but contain no valid ranged data, are marked as having been acquired, but no changes to the underlying models and the respective improvement calculations are made.
4) The calculation of the metrics used in the evaluation are then performed, and the resulting values are recorded. The duration of the evaluation metric calculation times that are not integral to the core algorithms (point insertions, and improvement calculations) are not recorded, as these are calculated primarily for the evaluation of the techniques.
5) The point selection process then chooses the next location to acquire data from, and the duration of the execution time of the point selection process is recorded.
6) Steps 3), 4), and 5) repeat until there are no more points left to insert into the models.

Note that actual range data acquisition time is considered to take zero times for the purposes of these evaluations. This is due to the acquisition of the scenes, which were fully captured and then stored in advance, are used to mimic a random access sensor that has identical characteristics to the one used to acquire the data. Furthermore the methods that are presented in Chapter 3 are designed to be independent of sensor acquisition technologies, which are characterized by different parameters, including speed of acquisition, and is not a factor directly considered in this thesis.

The metrics that are used for the evaluation of the techniques proposed in Chapter 3 include the surface mesh estimated improvement, the surface mesh mean estimated improvement, the surface mesh estimated improvement error, the peak signal to noise ratio (PSNR), the probabilistic occupancy grid estimated improvement, the probabilistic occupancy grid mean

estimated improvement, and the probabilistic occupancy grid estimated improvement error. A description of the metrics, and how they are obtained, are in the list below.

1) The *surface mesh estimated improvement* is the estimated surface mesh improvement at the location of the selected point, and is obtained from the surface mesh improvement map. This value is calculated intrinsically by the surface mesh improvement technique. This metric should decrease as more points are added to the surface mesh model, due to more knowledge being included leading to less potential improvement. This being said, there is expected to be localized increases in estimated improvement due to the discovery of additional features and discontinuities, but the overall trend should exhibit diminishing returns of the estimated improvement with respect to the number of points inserted into the surface mesh model. When comparing selection processes using this metric, a more rapid decrease in the trend towards the minimum implies better performance in regards to accurately representing the surface details.
2) The *surface mesh mean estimated improvement* is the mean of the entire surface mesh improvement map. This value is calculated externally to the core surface mesh improvement technique, which means that additional time is required to calculate this metric. This metric should exhibit similar trends to the surface mesh estimated improvement metric, with less peaky behaviour due to the smoothing effect that the calculation of the mean provides. As with the surface mesh estimated improvement, when comparing selection processes using this metric, a more rapid decrease in the trend towards the minimum implies better performance in regards to accurate scene representation.
3) The *surface mesh estimated improvement error* is the difference between the actual improvements a selected location provides when actually measured points are integrated into the model versus the estimated improvement from the potential improvement map for that location as computed in Chapter 3. The actual improvement is calculated externally from the core surface mesh improvement technique, and is calculated by summing the differences in the VMR before and after a point has been inserted into the surface mesh model. This metric should start off relatively high and decay as the accuracy of the scene improves – or in other words

the estimate becomes more reliable as additional points are added into the model. A better selection process will decrease this error more rapidly than others.

4) The *peak signal to noise ratio* is calculated based on the surface mesh estimated depth maps, which are produced intrinsically by the surface mesh improvement technique. This is accomplished by storing the estimated depth maps, after each point has been selected and inserted into the surface mesh model, into a losslessly compressed video file. In a post-processing analysis pass of the produced data, the PSNR is calculated based on the last frame of the resulting video file, which corresponds to the depth map when all points are inserted into the model, and the current frame of the video file using eq. (5.1), where $z(i)$ is the actual depth at location i as defined in the final depth map, and $\hat{z}_n(i)$ is the estimated depth after the n^{th} point has been inserted at location i. For the purposes of this thesis, the constant term of the traditional image based PSNR (peak term) is omitted due the range of measurements for each sensor being different, making it more difficult to compare curves between sensors on the same scale. When PSNR is mentioned in this thesis, the modified PSNR is used unless otherwise explicitly mentioned. The behaviour of the PSNR will increase to its maximum as more points are added to the surface mesh model due to the estimated depth map approaching the actual depth map. Outliers will cause the PSNR to be lower than expected. When using this metric to compare selection processes, a larger PSNR for a given number of points implies that the estimated depth map better represents the actual true depth map after only that number of points have been inserted into the model, which also means the selection process performs better, as fewer points need to be acquired for the model to reach a certain level of adequacy.

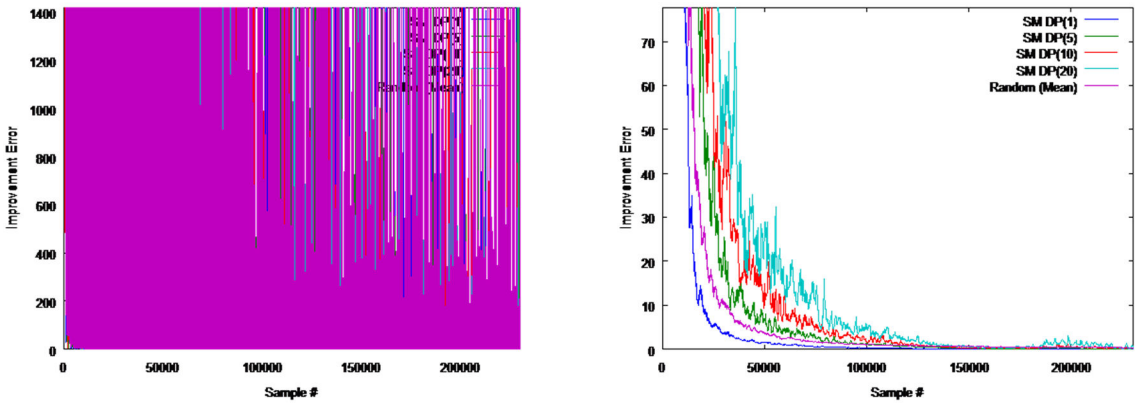
$$PSNR(n) = -10 \cdot log_{10} \sum_{\forall i} \left(z(i) - \hat{z}_n(i)\right)^2 \tag{5.1}$$

5) The *probabilistic occupancy grid estimated improvement* is the estimated probabilistic occupancy grid improvement at the location of the selected point, and is obtained from the probabilistic occupancy grid improvement map. This value is calculated intrinsically by the probabilistic occupancy grid improvement technique and it should converge to the final value as more points are inserted into the

probabilistic occupancy grid model, although due to negative improvement there will be peaks and valleys. A point selection process that better favours coverage will have this parameter stay at a higher value longer than other techniques, followed by a rapid decrease to a minimum improvement level.

6) The *probabilistic occupancy grid mean estimated improvement* is the mean of the entire probabilistic occupancy grid improvement map. This value is calculated externally to the core probabilistic occupancy grid improvement technique. A point selection process which favours coverage, will cause this metric to decay more rapidly to its minimal value, and then slowly rise to the final value, when compared to other techniques.
7) The *probabilistic occupancy grid estimated improvement error* is the difference between the actual improvements a selected location provides versus the estimated improvement from the improvement map for that location. The actual improvement is calculated externally from the core probabilistic occupancy grid improvement technique, and is calculated by summing the differences of the probability of occupancy of the surface of highest probability before and after a point has been inserted into the surface mesh model. For a point selection process that favours coverage, this value should be low for a long period of time, followed by an increase as coverage is approaching completion and more points along transitions where the estimates may be wrong are acquired.

In order so that the trends of the results are put into evidence, they are plotted after a median filter is applied. The size of the median filters used are 1% of the number of samples in the scene. This is performed since in evaluating the selection processes the trends are more important than the instantaneous data, since it shows the overall tendency in the evolution of the resulting metric. A median filter is used to ensure that the trends are not lost due to the large variation in obtained results, as is illustrated in Figure 5.1 a) where a sample of raw data is plotted, versus Figure 5.1 b) where the trends are put into evidence after a median filtering is applied on the raw data.



a) b)

Figure 5.1 - Improvement error for scene 1, a) raw data plot, and b) median filtered plot

Due to the behaviours of the surface mesh improvement based metrics, they provide an insight as to how well a particular selection process is in terms of accurately representing features within the scene. The behaviours of the probabilistic occupancy grid improvement based metrics, on the other hand, are useful in determining how well a selection process favours coverage. Furthermore, PSNR provides information as to how well the estimated surface mesh representation of the scene matches the complete representation of the scene, and is useful for measuring the quality of representation for a given number of points inserted into the surface mesh model.

Additional qualitative based evaluations of accuracy and coverage are performed, which include showing the progression of the acquisition of points for the selection processes for the surface mesh based maps (depth and improvement) and the probabilistic occupancy grid based maps (depth corresponding to the surface of highest probability, probability of occupancy corresponding to the surface of highest probability, and improvement).

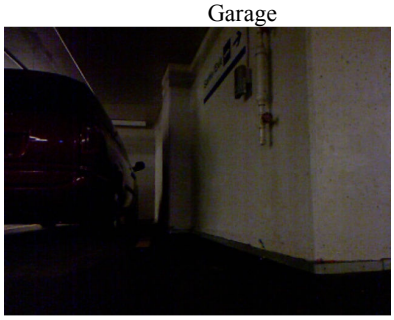
5.2 Surface Mesh Improvement Based Selection Process Evaluation

To perform the evaluation of the surface mesh improvement based selection process the random point selection process is used as a baseline comparison. The multi point selection process, $DP_1'(imp_{SM}, f, N)$ defined in eq. (3.84), is being used, with $f = 0$, and $N = \{1,5,10,20\}$. Furthermore the datasets of the Scenes 1-6 detailed in Appendix C are used, and briefly described below. Pictures of these scenes are reproduced in Figure 5.2 to support the evaluation.

a) Scene 1: Pillars in a Parking Garage

b) Scene 2: Corner in a Parking Garage

c) Scene 4: Cluttered Corner in a Parking Garage



d) Scene 4: Car and Walls in a Parking Garage

e) Scene 5: Pay Machine and Pillars in a Parking Garage

f) Scene 6: Cluttered Scene in a Lab

Figure 5.2 - RGB pictures of the six scenes used in the analysis for the surface mesh improvement selection process

Scene 1 (Figure 5.2 a), see Section C.1 for more details) consists of pillars at two depths in a parking garage and is acquired with the Kinect. This scene has cylindrical objects at different depths, as well as a rectangular box attached to the closest pillar and regions where the depth is beyond the operational limit of the Kinect. The second scene (Figure 5.2 b), see Section C.2 for more details), is an empty corner in a parking garage. What makes this scene interesting is the interior 90° where the walls meet the pavement. Scene 3, (Figure 5.2 c), see Section C.3 for more details), contains the same corner as Scene 2, but has some additional traffic cones and a filled recycling box providing more features at different depths in order to provide increased scene complexity. Scene 4, shown in Figure 5.2 d), (see Section C.4 for more details), consists of a car beside a wall in a parking garage. This scene has large areas where the Kinect could not acquire range data, in addition to surfaces that are nearly perpendicular to the viewer (the rear of the car, and part of the wall), and nearly parallel to the viewer (the side of the car and side wall). Scene 5, shown in Figure 5.2 e) (see Section C.5 for more details), shows some pillars surrounding a pay machine within a parking garage. This scene has large segments that are further away than what the Kinect can perceive, in addition to having cylindrical curves that are smaller than those presented in Scene 1, and the pay machine which is rectangular. Scene 6, shown in Figure 5.2 f) (see Section C.6 for more details), shows a cluttered scene within a laboratory environment. This

scene has many different surfaces at various depths and orientation. The black fender, in large part, is not perceived by the Kinect, the wooden pallet has 6 concave intrusions into its surface, in addition to channels connecting them, the mock car door has small features (handle and 3 protrusions), the small diameter cardboard cylinder, and the relatively flat boxes make this a relatively complex scene.

The first subsection presents the quantitative analysis of the surface mesh improvement based selection process versus the random point selection for the chosen scenes. The second subsection provides a qualitative analysis illustrating the progression of the surface mesh and probabilistic occupancy grid models as guided by the single point surface mesh improvement based selection process for the selected scenes. The third subsection provides an analysis on CPU execution time for each part of the complete selective sensing methodology, for each of the chosen scenes. Finally, the last subsection provides a comparison of the surface mesh improvement measure against the neural gas method [6,7], briefly described in Section 2.1.1, in determining regions that should be acquired in higher resolution.

5.2.1 Quantitative Analysis

This subsection presents and analyses the performance obtained with the proposed surface mesh improvement selection process against that of the random point selection process. The metrics being employed for the evaluation are the surface mesh mean estimated improvement, surface mesh estimated improvement error, the probabilistic occupancy grid mean estimated improvement, and the probabilistic occupancy grid estimated improvement error. The effect of the surface mesh improvement based selection process and the random point selection process on the probabilistic occupancy grid improvement measure based metrics are being used, as these provide insights on how well each selection process performs in regards to driving acquisition towards coverage of the scene, while the surface mesh improvement based metric provides insights on how well each selection process performs in regards to driving acquisition towards accuracy of the scene.

Graphs are presented which show the progression of each metric, based upon the locations selected by the surface mesh improvement based selection process and by the random point selection process. As most of the dynamic behaviour of the surface mesh improvement metrics takes place prior to 25000 points being inserted into the model, not only are the

graphs detailing the behavior for all points being inserted, but also graphs which contain the results truncated to contain only the first 25000 points are also included for a better visualization as to what occurs early on in the process. In all cases, the vertical range of the plots has been adjusted such that the main characteristics are better observed. For each set of results, the plots are labelled with the following convention:

a) Surface mesh mean estimated improvement for all the points.
b) Surface mesh mean estimated improvement for only the first 25000 points.
c) Surface mesh estimated improvement error for all the points.
d) Surface mesh estimated improvement error for only the first 25000 points.
e) Probabilistic occupancy grid mean estimated improvement for all points.
f) Probabilistic occupancy grid estimated improvement error for all points.

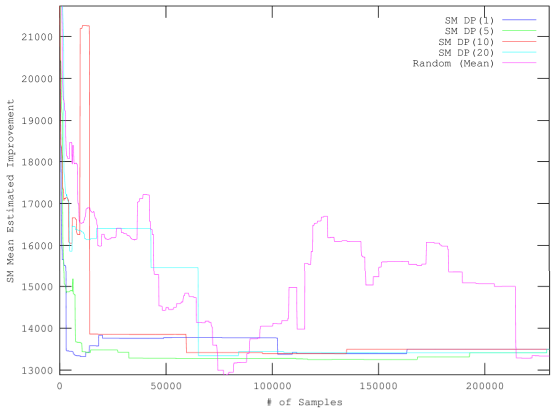
The legends on the plots referring to *SM DP(N)*, as exemplified in Figure 5.1, represent the surface mesh based improvement measure selection process with N points, where $N = \{1,5,10,20\}$ points are selected and inserted into the improvement measures at each iteration, that is collected in a single scan before the improvement map is updated. These N points belong to N separate triangles which have the highest improvement measures associated with them, as described in Section 3.3.1. The last entry in the legend, *Random (Mean)*, refers to the mean value of the corresponding metric calculated after the execution of 10 runs when using the random point selection process. Results for Scenes 1, 3, and 6 are presented in this section, the remainder for Scenes 2, 4, and 5 are presented in Section D.1.

5.2.1.1 Analysis of Scene 1

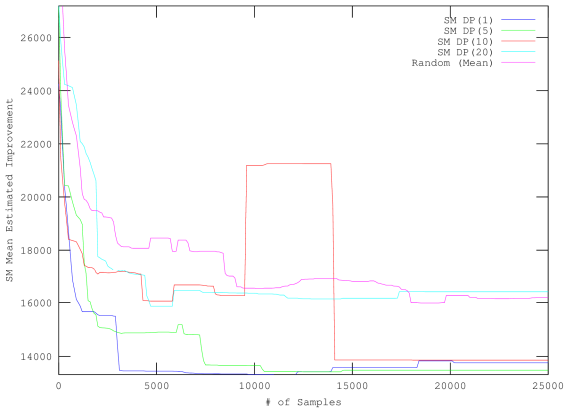
Scene 1 (see Section C.1) is the first scene considered to evaluate the algorithms against. This scene has cylindrical objects at different depths, as well as a rectangular box attached to the closest pillar and regions where the depth is beyond the operational limit of the Kinect. Figure 5.3 a) and b) shows that the surface mesh mean estimated improvement when using the surface mesh improvement based selection process (lines in blue, green, red, and cyan) performs better than random sampling (line in magenta), as these curves have less variation, and converge to their minimums with fewer points selected. Also, as evidenced in Figure 5.3 b), better performance is achieved when only 1 to 5 points are added at each iteration, as the curves that correspond to these scenarios, namely SM DP(1) in blue and SM DP(5) in green

decay sooner to the minimum mean improvement level than the other curves. The bumpy behaviour demonstrated by the SM DP(10) curve in red and the SM DP(20) in cyan occurs due to the larger number of points selected at each iteration providing more coverage of many different features as opposed to only along edges that provide the most improvement as occurs in SM DP(1) and SM DP(5), thereby revealing more regions of larger variation at once, instead of gradually. The random selection process curve in magenta exhibits the bumpy behaviour due to the larger coverage exposing the variation along the surfaces, even some caused by the noise of the Kinect. The plots showing surface mesh estimated improvement error in Figure 5.3 c) and d) indicate the estimated improvement value converges rapidly to the actual improvement that a particular location provides, and it achieves this prior to 5000 points inserted into the model. The two curves representing the SM DP(1) and random, which only select a single point at each iteration, provide more reliable estimates of improvement than the multipoint methods, as revealed by close inspection of Figure 5.3 d). It should be noted that SM DP(1) provides better performance than the random point selection, and indicates that the error in the estimation of the improvement, and thereby the scene, is reduced by acquiring fewer points, which is logical as the surface mesh improvement metrics gauge accuracy of representation of the scene, and random point selection favours broader coverage. Figure 5.3 e) graphing the probabilistic occupancy grid mean improvement indicates that the random point selection, since it has a much lower minimum than that of the surface mesh improvement based selection processes, greatly favours coverage, with maximum coverage occurring after approximately 50000 points have been inserted. The surface mesh improvement based selection processes on the other hand show little variation regardless of how many points are inserted at each iteration, with coverage being slowly increased as more points are being added to the models. This is to be expected as these selection processes focus on improving accuracy, which will gradually increase coverage as less dominant features are slowly acquired with higher accuracy. Finally, Figure 5.3 f) shows that the single point techniques have less probabilistic occupancy grid estimated improvement error than the multipoint techniques. This is due to the greater difference between the estimated and actual improvement caused by the addition of many points at a time. As can be observed, the random point selection process yields a

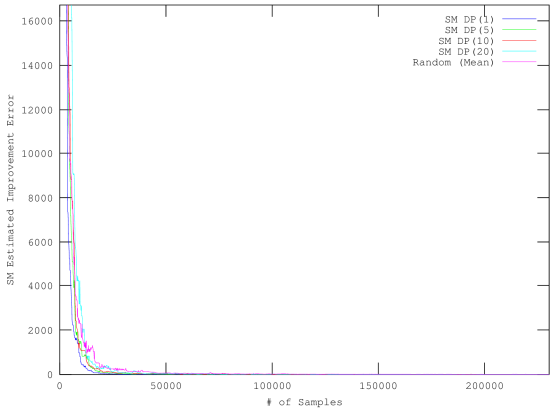
lower error than the surface mesh improvement based selection process, which is to be expected as random point selection favours coverage.



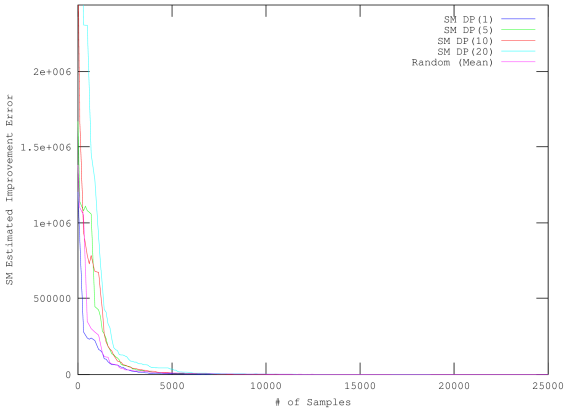
a) Surface mesh mean estimated improvement over all points



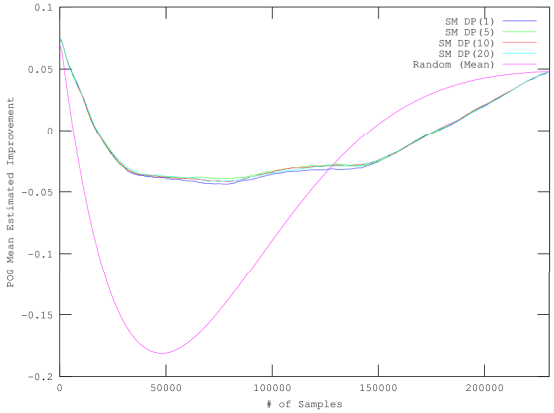
b) Surface mesh mean estimated improvement over 25000 points



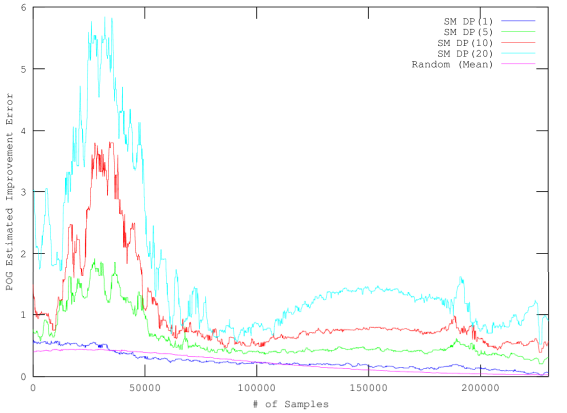
c) Surface mesh estimated improvement error over all points



d) Surface mesh estimated improvement error over 25000 points



e) Probabilistic occupancy grid mean estimated improvement over all points

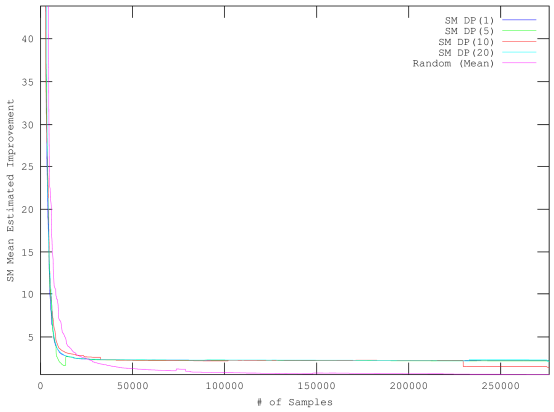


f) Probabilistic occupancy grid estimated improvement error over all points

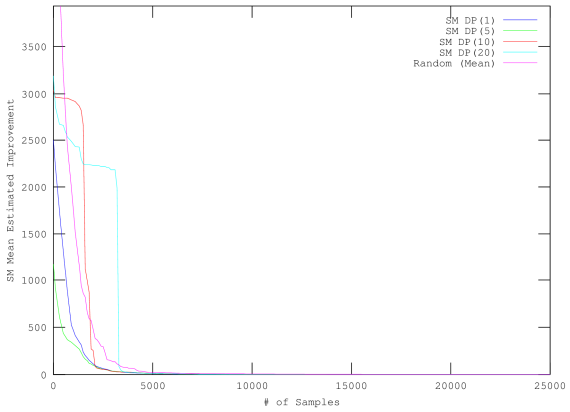
Figure 5.3 - Scene 1: mean improvement and improvement error trend graphs

5.2.1.2 Analysis of Scene 3

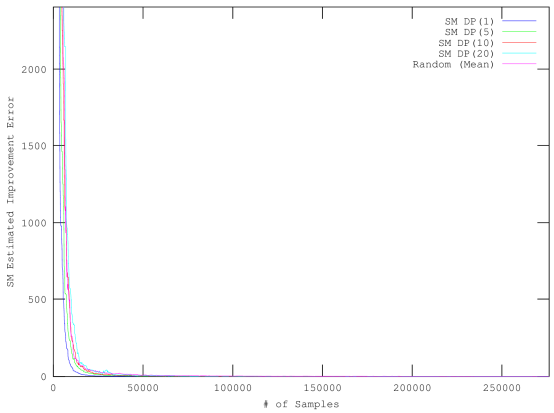
Figure 5.4 a) and b) show the surface mesh mean estimated improvement for Scene 3 (see Section C.3), which indicates that the surface mesh improvement based selection process converges to its minimum with fewer points inserted into the model than what occurs when using the random point selection process. The performance of the surface mesh improvement based selection process when N=5, performs better than N=1, with the other two, namely when N=10 and 20, performing much worse. This is likely due to their only being a few objects in the scene, and hence for the two latter cases, points that are being selected are modelling the noise as opposed to real objects. Figure 5.4 a) and b) puts into evidence the fact that convergence of this metric to the minimal values occurs after approximately 5000 points have been inserted into the model. Similarly, Figure 5.4 c) and d) indicate that the surface mesh estimated improvement error for the surface mesh improvement based selection process when N=1 and N=5 converges prior to random point selection, with the other two multipoint surface mesh improvement based selection processes performing worse for this metric. This is due to the previous observation that with more points being inserted per iteration, the underlying sensor noise from the Kinect is modelled, leading to slightly better coverage, but less focus on accuracy. Figure 5.4 e), which shows the probabilistic occupancy grid mean estimated improvement metric, puts into evidence the same trend that has been observed with Figure 5.3 e), namely that the random point selection process has better coverage than the surface mesh improvement based selection process, with maximum coverage occurring after approximately 50000 points have been inserted, and that the number of points being chosen by the surface mesh improvement based selection process at each iteration does not greatly affect the scene coverage. Figure 5.4 f) detailing the probabilistic occupancy grid estimated improvement error, indicates the same trend as previously observed, namely that the random point selection process produces better coverage, as the curve is lower valued than that produced from the surface mesh improvement based selection processes for most of the plot.



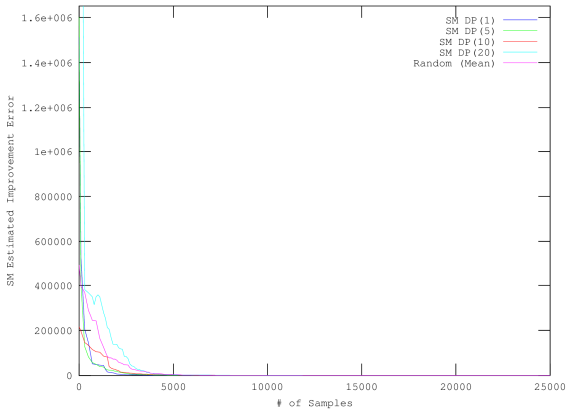
a) Surface mesh mean estimated improvement over all points



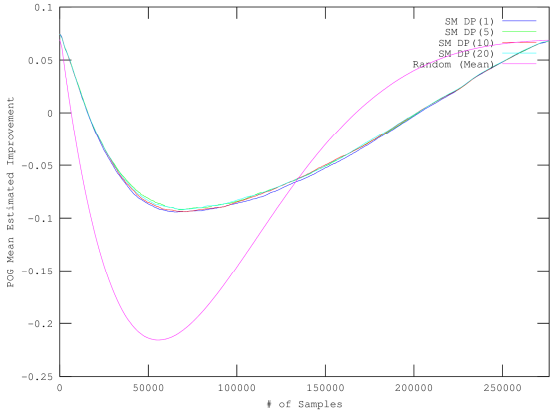
b) Surface mesh mean estimated improvement over 25000 points



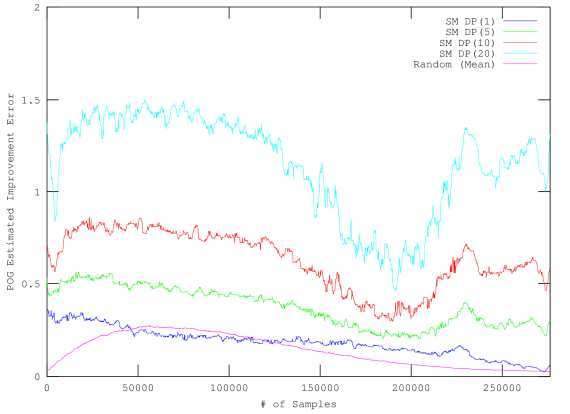
c) Surface mesh estimated improvement error over all points



d) Surface mesh estimated improvement error over 25000 points



e) Probabilistic occupancy grid mean estimated improvement over all points

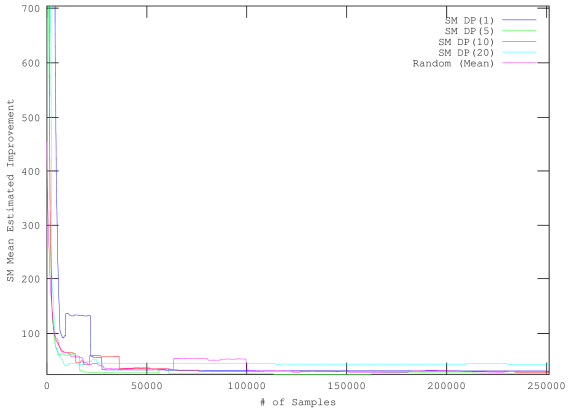


f) Probabilistic occupancy grid estimated improvement error over all points

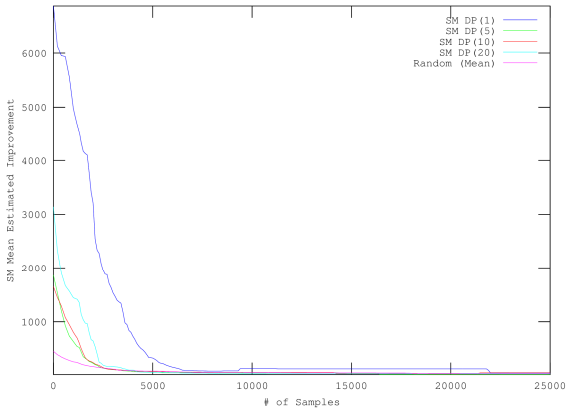
Figure 5.4 - Scene 3: mean improvement and improvement error trend graphs

5.2.1.3 Analysis of Scene 6

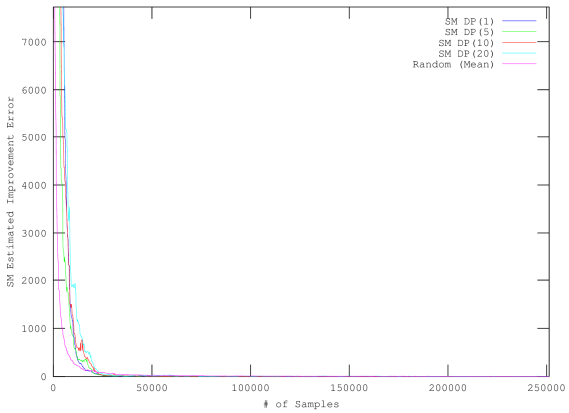
For the performance of the selection processes on Scene 6 (see Section C.6), Figure 5.5 a) and b) show that the surface mesh mean estimated improvement when using the surface mesh improvement based selection process, for all values of N tested, converges initially at a slower rate than random sampling, but reaches the minimum mean improvement level prior to random point selection, except for the case where N=1. This is due to the selection processes selecting consecutive points in a zigzag pattern, which will be put into evidence by Figure 5.18 in Section 5.2.2.3. Figure 5.5 c) and d), showing the surface mesh estimated improvement error, demonstrates that for this particular scene the random point selection process produces a lower error than does the surface mesh improvement based selection process. Figure 5.5 d) puts into evidence that although all curves converge to their respective minimums after approximately 10000 points have been inserted into the model, the case when N=1 performs worse than the other cases, with the case when N=5 performing the best out of the surface mesh improvement based selection processes. Figure 5.5 e) showing the progression of the probabilistic occupancy grid mean estimated improvement metric, demonstrates similar behaviour as exhibited in the previous scenes studied, namely that the minimum value of the curve produced using the random point selection process achieves a significantly lower value than that produced using the surface mesh improvement based selection processes, which indicates that as expected, the random point selection process provides better scene coverage than the surface mesh improvement based selection processes. Similarly, Figure 5.5 f) showing the probabilistic occupancy grid estimated improvement error, demonstrates the behaviour of the random point selection process curve being lower for most of the plot when compared to the surface mesh improvement based selection process curves, indicating that the random point selection process does indeed provide better scene coverage.



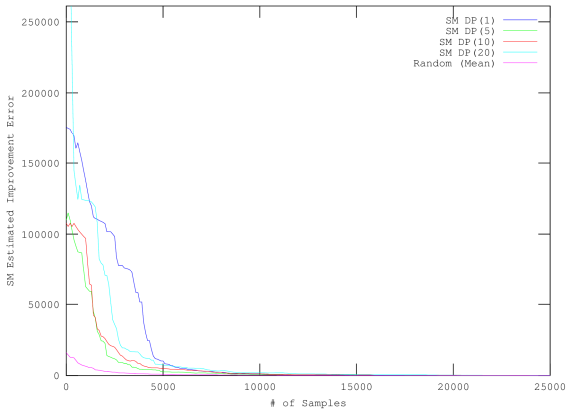
a) Surface mesh mean estimated improvement over all points



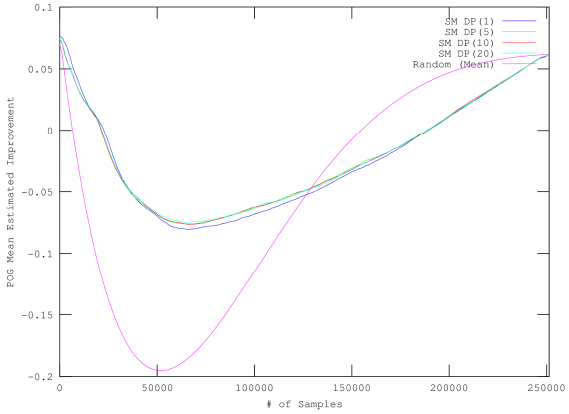
b) Surface mesh mean estimated improvement over 25000 points



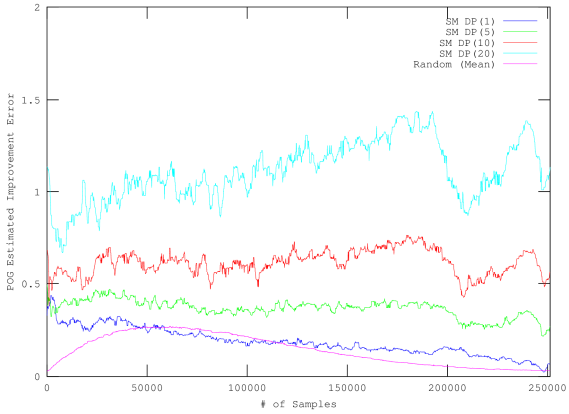
c) Surface mesh estimated improvement error over all points



d) Surface mesh estimated improvement error over 25000 points



e) Probabilistic occupancy grid mean estimated improvement over all points



f) Probabilistic occupancy grid estimated improvement error over all points

Figure 5.5 - Scene 6: mean improvement and improvement error trend graphs

5.2.1.4 Summary of the Quantitative Analysis

All six scenes studied (Scenes 1, 3 and 6 in Section 5.2.1, and Scenes 2, 4, and 5 in Section D.1) exhibit similar overall trends for Figure 5.3 - Figure 5.5 (and for Figure D.1 - Figure D.3). For the subfigures a), which is the graph of surface mesh mean estimated improvement containing all the points available from the respective point clouds, the general trend is a convergence to a specific level of improvement, which is the expected behaviour as previously discuss in Section 5.1. Subfigures b) show more of the initial transient in the graph of surface mesh mean estimated improvement as the analysis concentrates over the first 25000 points acquired. They emphasize that the rapid decrease of mean improvement from a high value to the level of convergence at between 5000 - 10000 points acquired and inserted into the model in most cases. This metric is noisier and requires a greater number of points to converge for scenes that have large regions that cannot be perceived by the sensor, such as Scenes 1 and 5 as put into evidence by Figure 5.3 b) and Figure D.3 b) respectively. As shown in Figure D.1 a) and b) and Figure 5.5 a) and b) for Scenes 2 and 6 respectively, the random point selection process outperforms the surface mesh improvement based selection processes for the surface mesh mean estimated improvement, due to a zigzag pattern being performed early in the acquisition procedure, which will be discussed in more detail in Section 5.2.2.4. The other scenes show that the surface mesh improvement based selection process performs better than the random point selection process for this metric. Subfigures c) and d) show the convergence to zero for the surface mesh estimated improvement error, with d) showing the initial rapid decrease in improvement error and with convergence occurring for this metric at between 5000-10000 points acquired, with more complex scenes, such as Scene 6, requiring more points than less complex scenes. Furthermore, the surface mesh improvement based selection process outperforms the random point selection process in this metric, except for Scenes 2 and 6, as can be observed in the more rapid convergence. Except for these two scenes, the additional selection of more points per iteration slows down the convergence, as is observed for the cases when N=5, 10, and 20. This is a result of adding more points, as when adding more points at a time creates a larger difference between the estimated and actual improvement, thus increasing this error. Subfigures e) illustrate similar curves with a bathtub shape, where the probabilistic occupancy grid mean estimated improvement starts off high, then decreases to a minimum,

and then increases to the final value. The random point selection process generates a more dramatic drop to the minimum value, which occurs after approximately 50000 points have been inserted, while the surface mesh improvement based selection processes generating shallower dips, with no significant difference between the curves based on the number of points inserted with each iteration. This indicates that the random point selection process favours scene coverage than what occurs in an approach that favours accuracy, such as implemented by the surface mesh improvement based selection process. The last subfigures f), which are the probabilistic occupancy grid estimated improvement error graphs, show consistently that the random point selection process has a lower value over most of the plot, than the surface mesh improvement based selection process, which is indicative of the random point selection process producing better scene coverage. Furthermore, as more points are inserted per iteration, the error increases. This is due to there being larger differences between the actual improvement and the estimated improvement that occurs when more points are inserted per iteration yielding a larger error.

In summary, based on the evaluation metrics used in this section, the surface mesh improvement based selection process performs better than the random point selection process at accurately capturing scene accuracy. This is illustrated through the surface mesh mean estimated improvement, as well as the surface mesh estimated improvement error. In some scenes, the random point selection process appears to have an initial advantage over the surface mesh improvement based selection process in terms of surface mesh mean estimated improvement (subfigures a) and b) of Figure D.1 and Figure 5.5), where a zigzag scanning pattern is established, but even is these scenarios the surface mesh improvement based selection process eventually converges to the minimum value. The addition of more points at a time tends to raise the surface mesh estimated improvement error, but provides more consistent speed in convergence to the overall minimum improvement level, mainly due to scanning in multiple regions at the same time and avoiding the pitfalls of an initial subsampling that may have missing points on the outside edge of the field of view, as long as not too many points are acquired at a time. N=5 was the most consistent out of the multiple point results. The probabilistic occupancy grid mean estimated improvement and the probabilistic occupancy grid estimated improvement error show that the random point

selection process greatly favours improved coverage when compared against the surface mesh improvement based selection processes.

Moreover, the improvement measures demonstrate that they perform well and are stable, namely that they converge and exhibit consistent behaviour across a wide variety of scenes and circumstances. Of particular interest for the application of selective sensing for reducing the number of points needed to be acquired to represent a scene, convergence of the surface mesh mean estimated improvement measures occurs at between 5000 and 10000 points inserted into the model, indicating that most of the features of the scene have been accurately captured. The qualitative behaviour will be analysed in the following section.

5.2.2 Selection Process Progression

The surface mesh improvement based selection process will be evaluated qualitatively in this section. As discussed previously in Section 5.2.1, the majority of the decrease in surface mesh mean estimated improvement and surface mesh estimated improvement error occurs after 5000 points being selected and inserted into the model. In order to show the progression of the selection process within this critical region, the improvement maps as well as the depth maps corresponding to when the first 1000, 2000, 3000, 4000, and 5000 points, and after all points available are inserted into the surface mesh and probabilistic occupancy grids, are presented. The results contained in this section are produced using the surface mesh improvement based selection process defined in eq. (3.83) for each of the six scenes used in the evaluation of Section 5.2.

In the depth maps, black pixels correspond to zero depth, while white pixels correspond to maximum depth (which is 5.0 m for this case using the Kinect), and gray pixels represent depths in between these two extremes (Figure 5.6 a). In the surface mesh improvement map, white corresponds to the pixels where the estimated improvement is highest, and black to where there is no estimated improvement, with gray corresponding to estimated improvement levels in between these two extremes (Figure 5.6 b). The probability of occupancy map corresponding to the surface of highest probability is white, that is where the probability of occupancy is sure ($P(Occ) = 1.0$), black where there is no probability of occupancy ($P(Occ) = 0.0$), gray where the probability of occupancy is unknown ($P(Occ) = 0.5$), and the other shades of gray for the appropriate values in between (Figure 5.6 c), The

legend corresponding to the artificial colouring in the probability occupancy grid improvement map is presented in Figure 5.6 d).

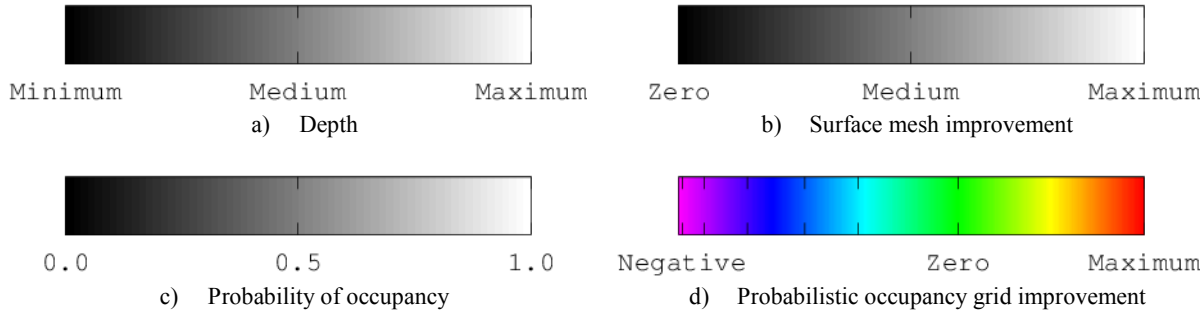


Figure 5.6 - Legends of colour vs. value for each metric studied

The evolution of the surface mesh improvement map, when using the surface mesh improvement based selection process with N=1, occurs by selecting the largest improvement in the entire map (white pixel) that corresponds to a location in the sensor field of view not previously acquired. As more points are acquired, many triangles are formed, and so in regions where there are larger depth transitions, and thereby higher improvement values, there will be many more smaller triangles. In scenes where there are large regions where data cannot be acquired (as in Scene 1, Figure 5.7 a)), there remain several large triangles with high improvement, as these have been attempted to be acquired, but since the sensor cannot acquire data in these regions that are beyond its operational depth of field, other high improvement locations within the field of view are selected instead. As more points are added to the surface mesh, the overall potential improvement values decrease, and so the improvement map should eventually move to a constant level of gray, except for large regions that cannot be acquired because of sensor's limitations. Furthermore as more points are selected and inserted into the model, transition regions in the surface mesh depth map become sharper, and there are less harsh distinct boundaries between triangles.

The evolution of the probabilistic occupancy grid, as driven by the surface mesh improvement based selection process, is also presented in this section. This evolution is presented for two reasons. Firstly, it will provide insight on how the two improvement measures will complement each other. Secondly, the regions that have been chosen for acquisition by the surface mesh improvement based selection process can be directly visualized by the evolution of the probabilistic occupancy grid improvement map through the

regions of negative improvement (in blue), which surrounds values that have been inserted already into the probabilistic occupancy grid. Since the maximum probability of occupancy and corresponding depth map demonstrate similar behaviours to the probabilistic occupancy based improvement map, they are only shown for the first dataset in Section 5.2.2.1, and from there on, these maps are not presented anymore. Furthermore, for the first scene, a comparison against the progression achieved by the random point selection is made.

The resulting progression and analysis for Scenes 1, 3, and 6 will be presented in this section. The remaining 3 scenes, namely Scenes 2, 4, and 5, will have their progression and analysis presented in Section D.2. After the results of each dataset are discussed, and overall summary will be made in the last subsection.

5.2.2.1 Scene 1 - Pillars in a Parking Garage

Figure 5.7 shows the progression of the surface mesh based improvement map for Scene 1 shown in Section C.1, while Figure 5.8 shows the corresponding depth map. Notice that at the beginning, in Figure 5.7 a), the larger surfaces of the ground and the large center pillar have lower improvement values assigned to them as evidenced by the darker gray triangles, and areas where there is no data or where there is a transition region have higher improvement values assigned to them, with the shading of these triangles being lighter gray to white. In all the subfigures, the region of highest improvement (white) located in the right third of the scene, corresponds to a region where there is no data available from the Kinect because the scene is beyond its depth of field, and it also happens to be a large depth transition area with the main central pillar of the scene. This region also coincides with the region of highest improvement (red in Figure 5.10 f)) for the probabilistic occupancy grid improvement map. The selection process, after determining that there is no acquirable data in the region surrounding the large bright white triangle, directs the acquisition of points along the transition regions where there are lighter gray triangles. This makes the depth transition regions become sharper and more distinct as more points are selectively added, as well as exploring the curvatures of the pillars, as evidenced in the depth map, Figure 5.8. Figure 5.10 shows the progression of the probabilistic occupancy improvement map, while the progression of the surface of highest probability of occupancy is illustrated in Figure 5.11 (depth of the surface) and Figure 5.12 (probability over the surface). Due to the small

number of acquisitions made at the times captured by subfigures a)-e), the probabilistic occupancy improvement map, Figure 5.10, shows large regions of maximum improvement (red), along with some small segments of negative (blue) and low improvement (green-yellow) along locations where points have been acquired and inserted into the probabilistic occupancy grid. Figure 5.10 f) also illustrates regions in green, which correspond to near zero potential improvement, due to the values being eventually inserted in these scanning locations exceeding the supported range of depths of the sensor (0.8m - 5.0m). This causes the probabilistic occupancy to map these scanning directions to be empty with near certainty ($P(occ)\sim 0.0$) as shown by the black regions in Figure 5.12 f), and hence offering very little room for improvement. By observing the progression between subfigures a)-e) in Figure 5.10, Figure 5.11, and Figure 5.12, the actual directed acquisition that results from the application of the proposed surface mesh improvement based selection process detailed in Section 3.3.1 can be observed. Note that after only 4000 points have been inserted, as can be observed in Figure 5.8 d), the main pillar is fairly accurately represented, as is the box attached to its top, which coincides to the quantitative observation that can be noted from the point of convergence of the surface mesh mean estimated improvement shown in Figure 5.3 b), and by the point of convergence of the surface mesh estimated improvement error shown in Figure 5.3 d).

A comparison of the surface mesh improvement based selection process and the random point selection process in terms of progression of the available depth maps can be made by comparing Figure 5.8 against Figure 5.9. The comparison of subfigures a)-e) between the two figures, shows that the central pillar becomes defined, with crisper edges becoming evident, earlier with the proposed surface mesh improvement based selection process, in Figure 5.8, than with the random point selection process, in Figure 5.9, where relative sharpness does not occur until about 28000 points (Figure 5.9 f)) have been randomly inserted. Note that the random point selection process, on the other hand, has better coverage elsewhere in the scene, as can be illustrated by the smaller triangles on the floor of the scene in Figure 5.9 a). But as these points are acquired where the scene is relatively flat, they contribute less to the overall knowledge about the accurate representation of the scene when compared to points acquired with the proposed surface mesh improvement based selection process that are closer to where edges and other depth features are located.

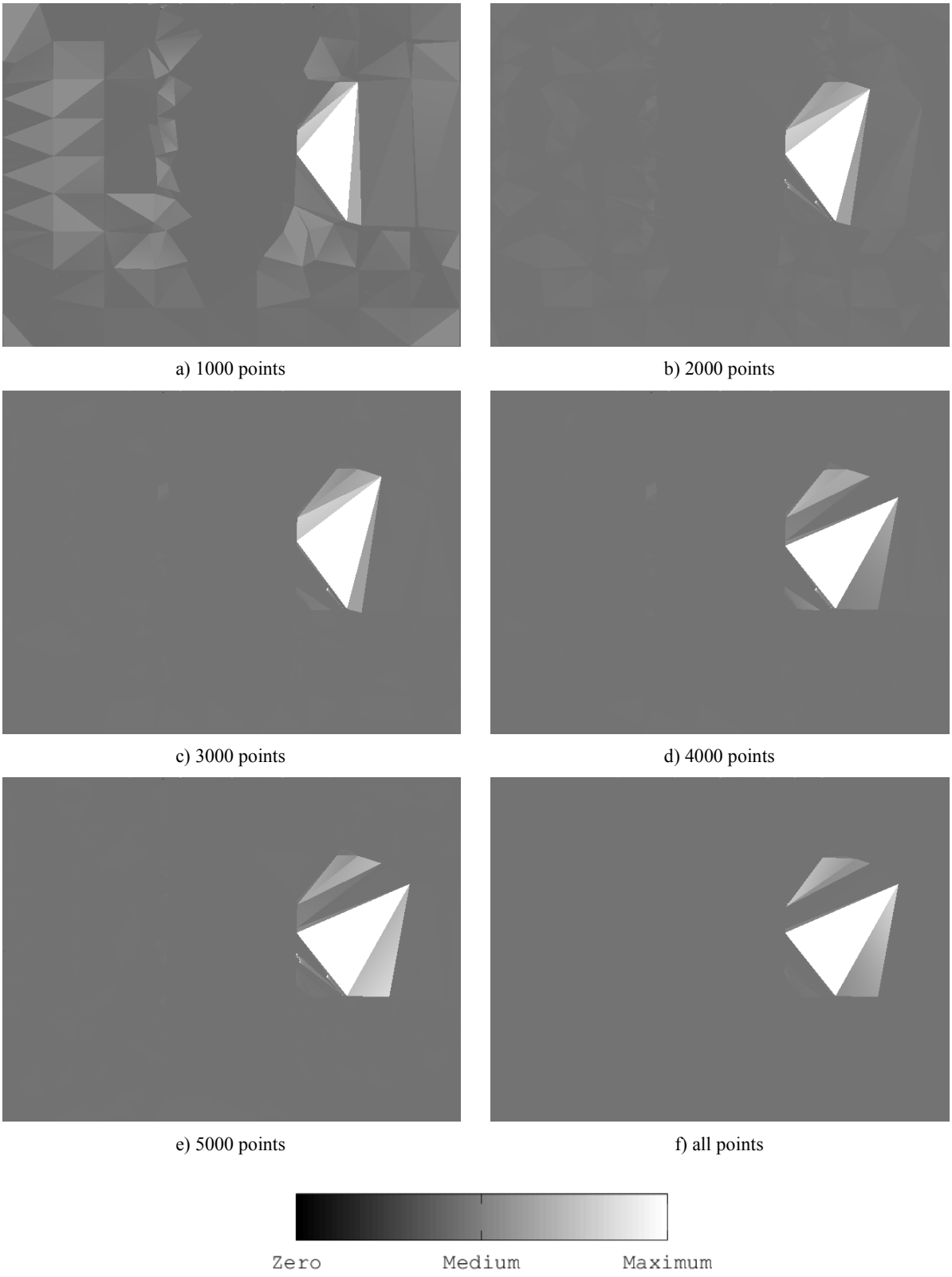


Figure 5.7 - Scene 1: surface mesh improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

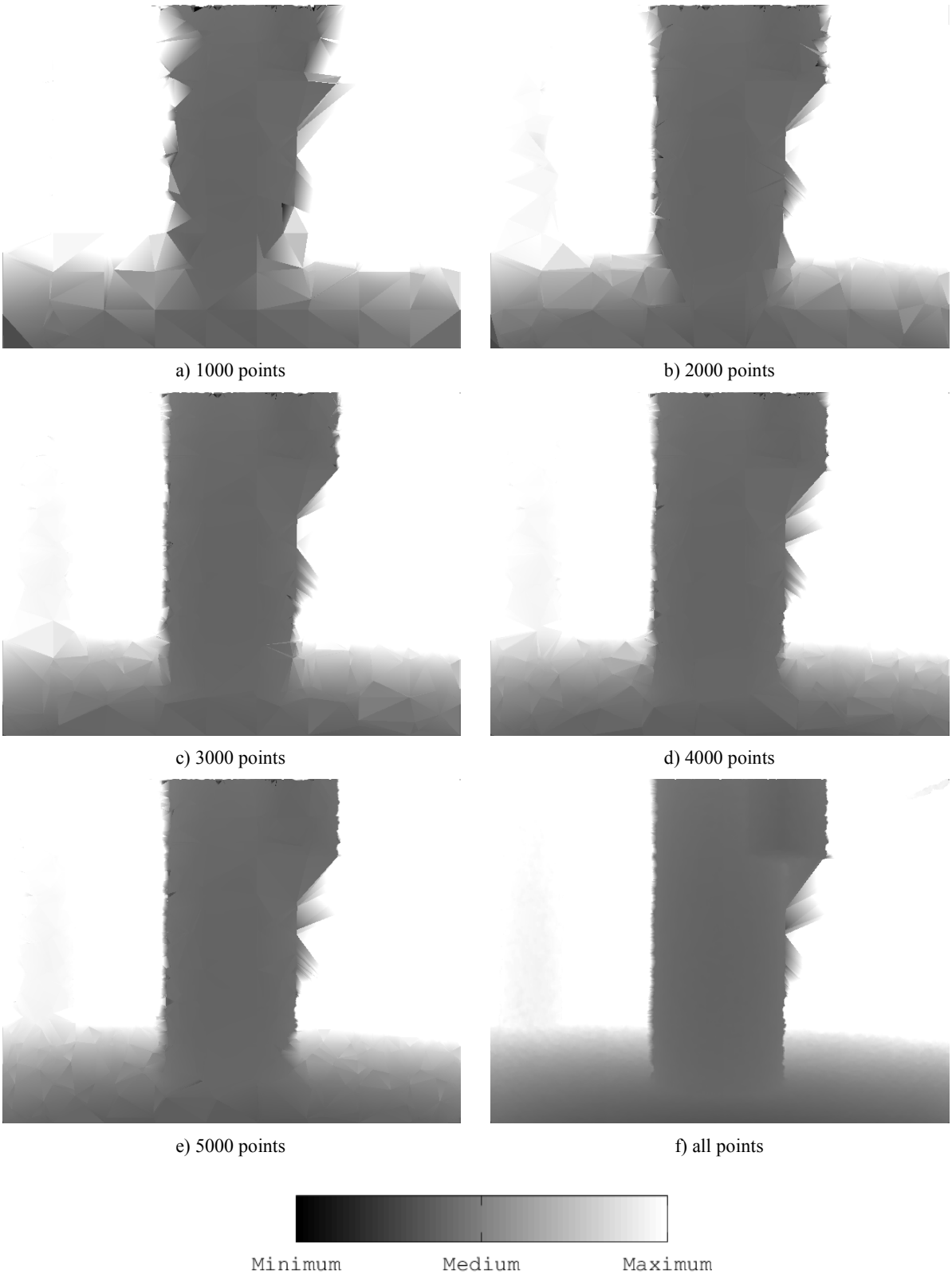


Figure 5.8 - Scene 1: surface mesh based depth map progression after different numbers of points are acquired using the surface mesh improvement based selection process

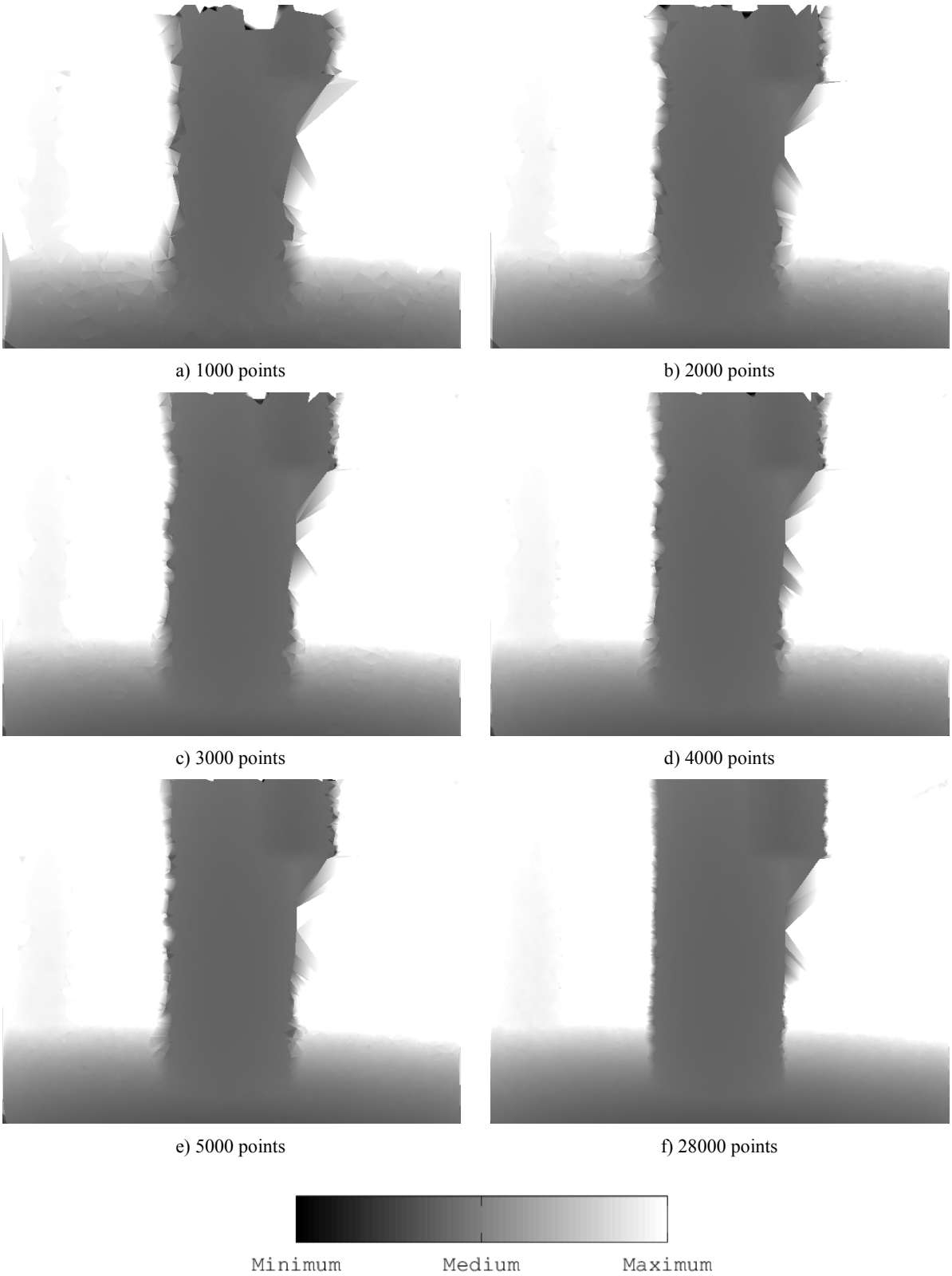


Figure 5.9 - Scene 1: surface mesh based depth map progression after different numbers of points are acquired using the random point selection process

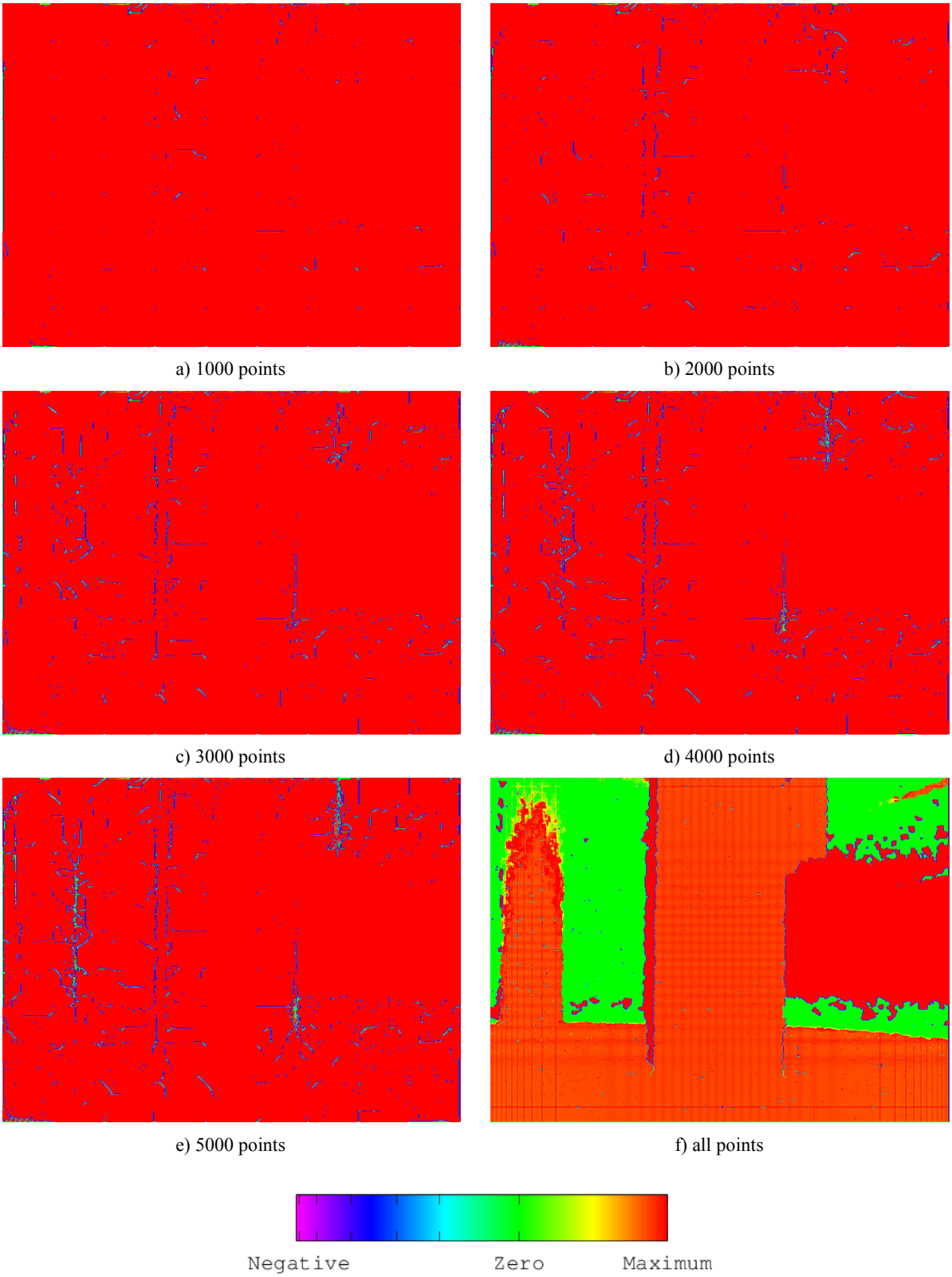


Figure 5.10 - Scene 1: probabilistic occupancy grid improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

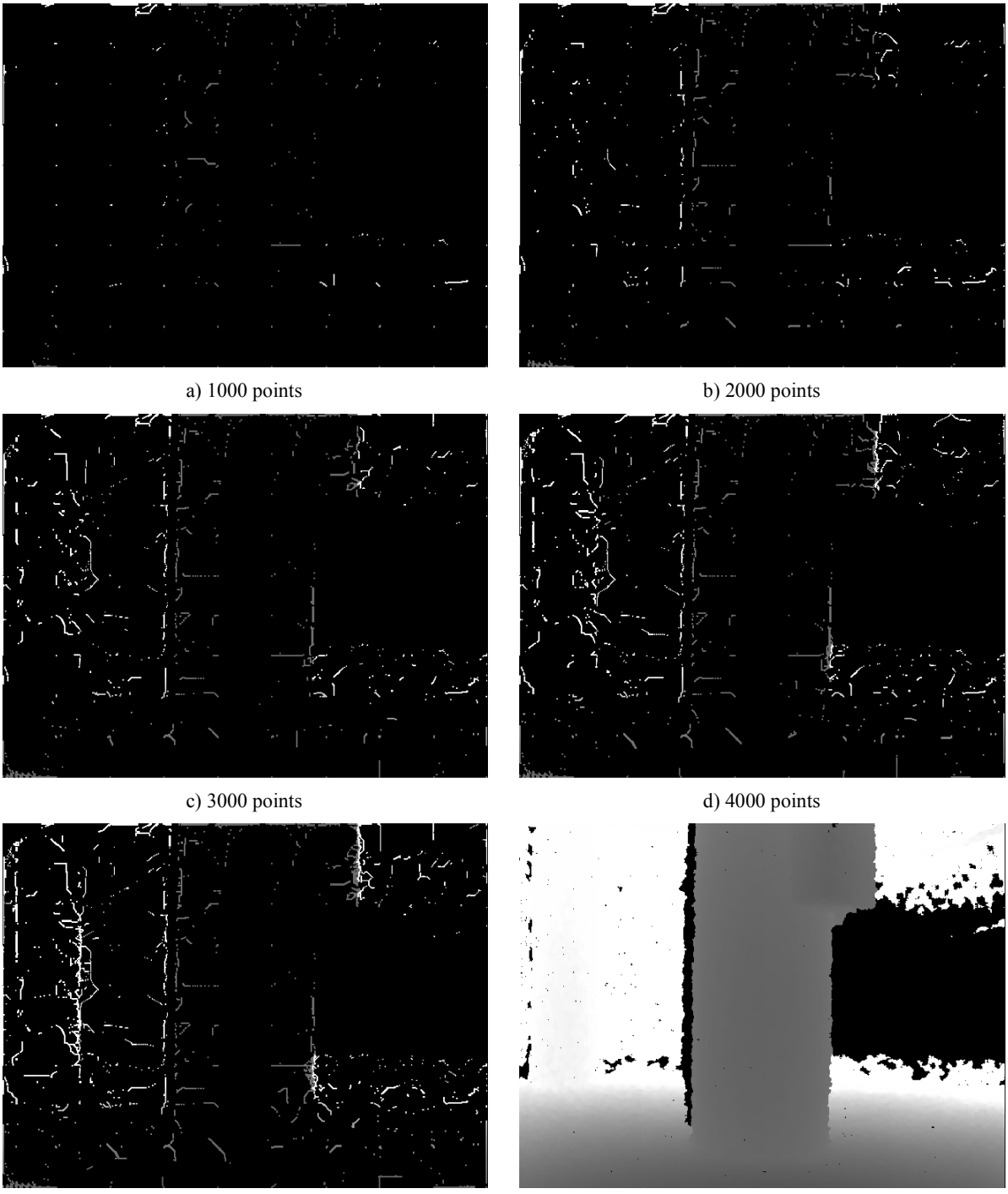
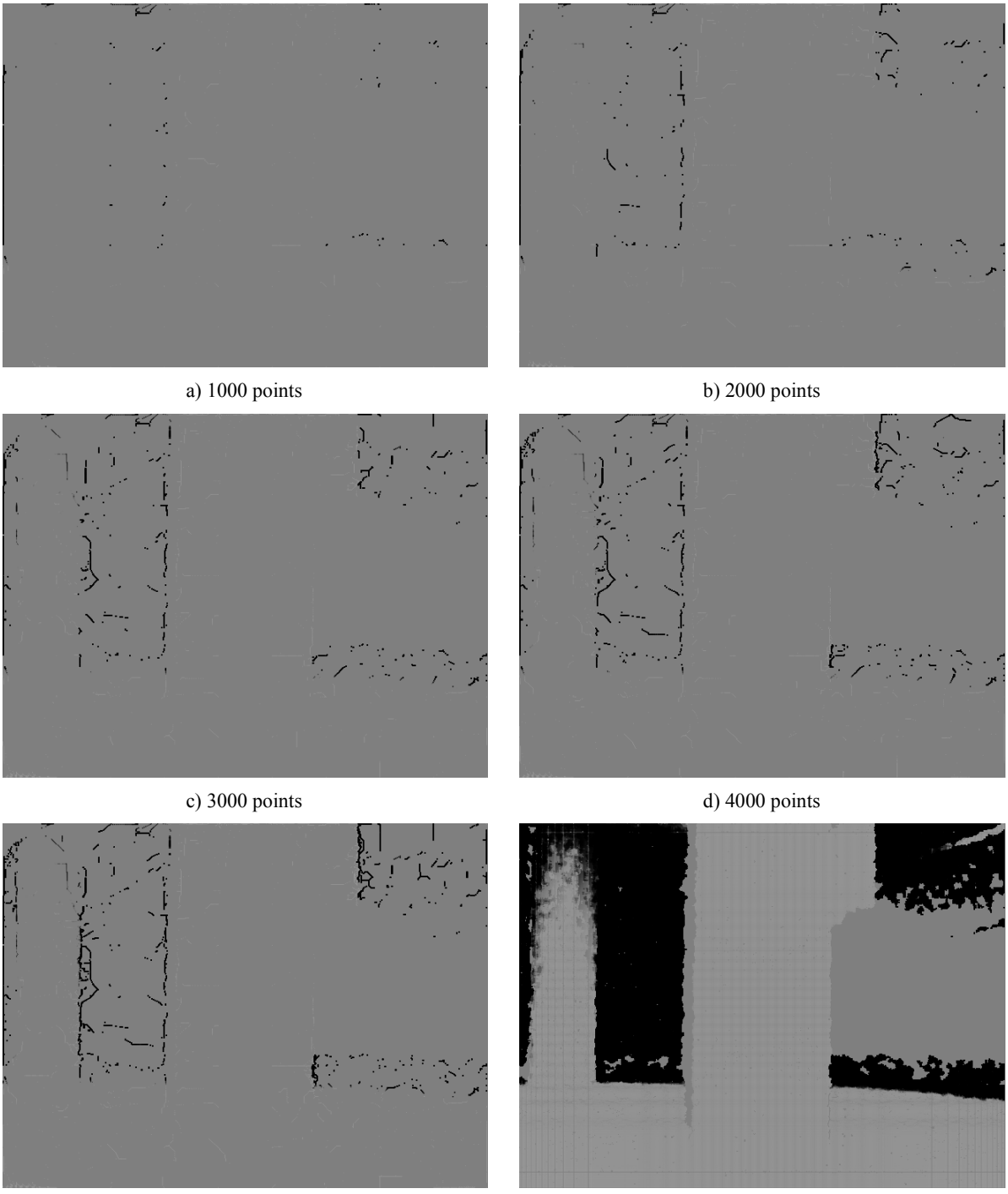


Figure 5.11 - Scene 1: probabilistic occupancy grid based depth map progression after different numbers of points are acquired using the surface mesh improvement based selection process



a) 1000 points

b) 2000 points

c) 3000 points

d) 4000 points

e) 5000 points

f) all points

Figure 5.12 - Scene 1: probabilistic occupancy grid based probability of occupancy map progression after different numbers of points are acquired using the surface mesh improvement based selection process

5.2.2.2 Scene 3 - Cluttered Parking Garage Corner

In Scene 3 of the cluttered parking garage (shown in Section C.3), the progression of the scene between Figure 5.13 a) and b) shows that the surface mesh improvement based selection process is guided to the transition region around the cones, as is illustrated by the bright white triangles that become progressively smaller. Figure 5.15 shows that the inherent behaviour of the proposed surface mesh improvement based selection process is to scan around these high depth transition objects (cones), and next around medium level transitions like the floor/wall and wall/wall interfaces before proceeding to the lower transition objects and features (the recycling bin against the wall). When the 5000^{th} point is inserted the scene has already acquired an accurate representation of the scene illustrated by comparing Figure 5.14 e) to Figure 5.14 f) when all available points have been inserted. Furthermore, this coincides with the point of convergence of the surface mesh mean estimated improvement for this scene and of the surface mesh estimated improvement error, as evidenced by Figure 5.4 b) and d) respectively. Note that the occlusion regions around the cones (where the Kinect IR projector and IR camera cannot both reach to collect the depth data) are labelled as regions of high improvement in both improvement maps as indicated in Figure 5.13 f) and Figure 5.15 f). Furthermore, the regions that contain the highly reflective white strips lay on the cones (the IR does not reflect well off of it) are labelled on the probabilistic occupancy grid improvement map (Figure 5.15 f)) as regions of high improvement potential, while the surface mesh based improvement measure interpolates well across this relatively small region, as evidenced by Figure 5.13 f) and Figure 5.14 f).

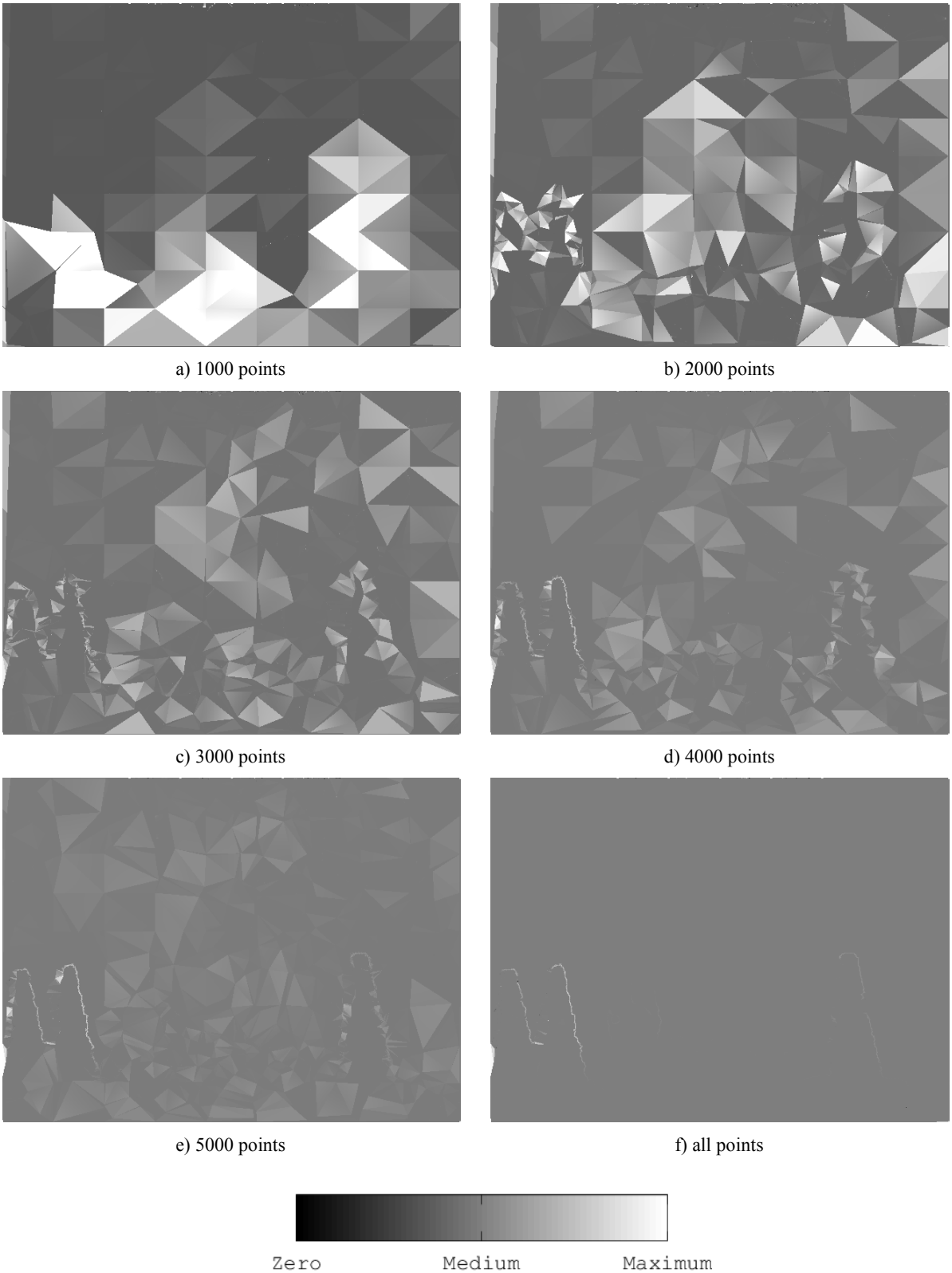


Figure 5.13 - Scene 3: surface mesh improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

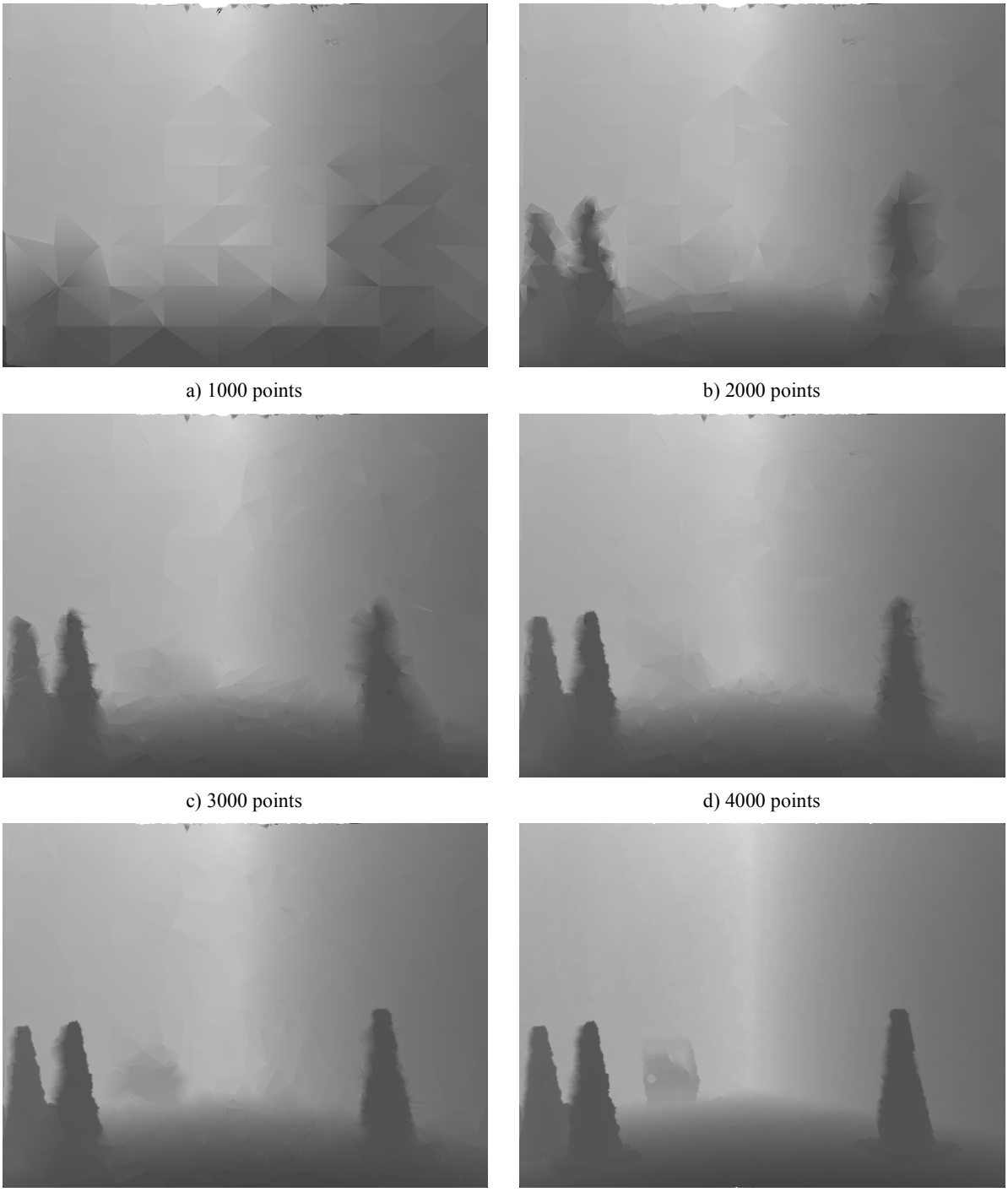


Figure 5.14 - Scene 3: surface mesh based depth map progression after different numbers of points are acquired using the surface mesh improvement based selection process

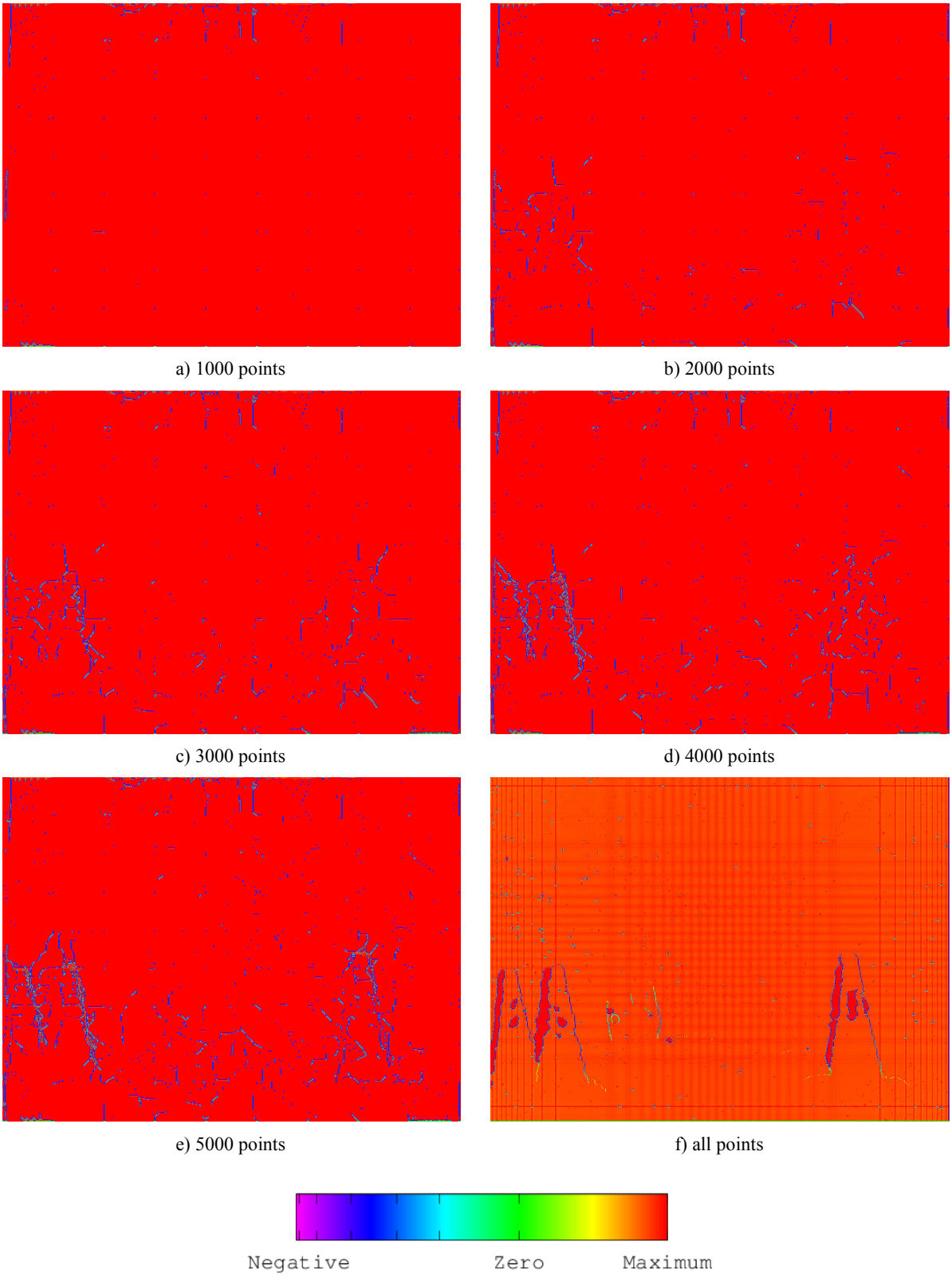


Figure 5.15 - Scene 3: probabilistic occupancy grid improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

5.2.2.3 Scene 6 - Cluttered Lab Scene

In this case, for Scene 6 shown in Section C.6, the surface mesh improvement based selection process produces point selections in a vertical zigzag scanning pattern in the bottom left region (Figure 5.18). This behaviour is due to not having enough points in the initial 10x10 subsampling in the far left of the scene, as evidenced by the large partial white triangles in Figure 5.16 a). This is reflected by Figure 5.5 b) in Section 5.2.1.3, the selection process where N=1 takes until after 5000 points to reach close to the convergence level of the surface mesh mean improvement graph. After performing this vertical zigzag scanning on the left hand side, the latter area is acquired at high resolution as observed in Figure 5.17 d). As shown in Figure 5.16 d) and e), when the zigzag acquisition pattern is finished, the remaining region of high improvement in the surface mesh improvement map (white and light gray) correspond to triangles spanning the transition regions around the palette, mock car door, wall-to-ceiling intersection, and car wheel well panel. This coincides well with the quantitative results presented in Figure 5.5 from Section 5.2.1.3, which show that the random point selection process initially progressed more rapidly towards the convergence point for the surface mesh mean estimated improvement metric than the single point surface mesh improvement based selection process.

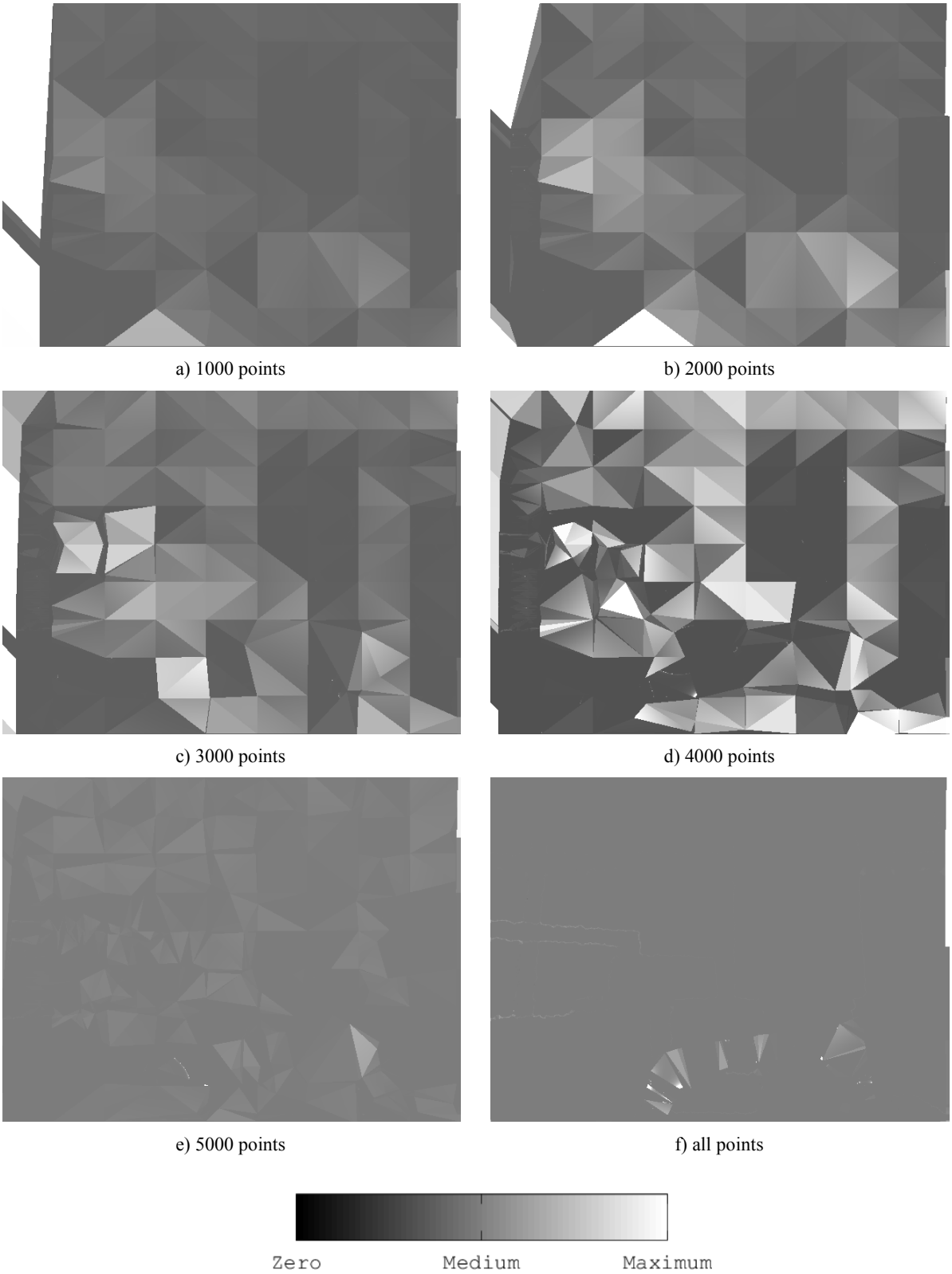


Figure 5.16 - Scene 6: surface mesh improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

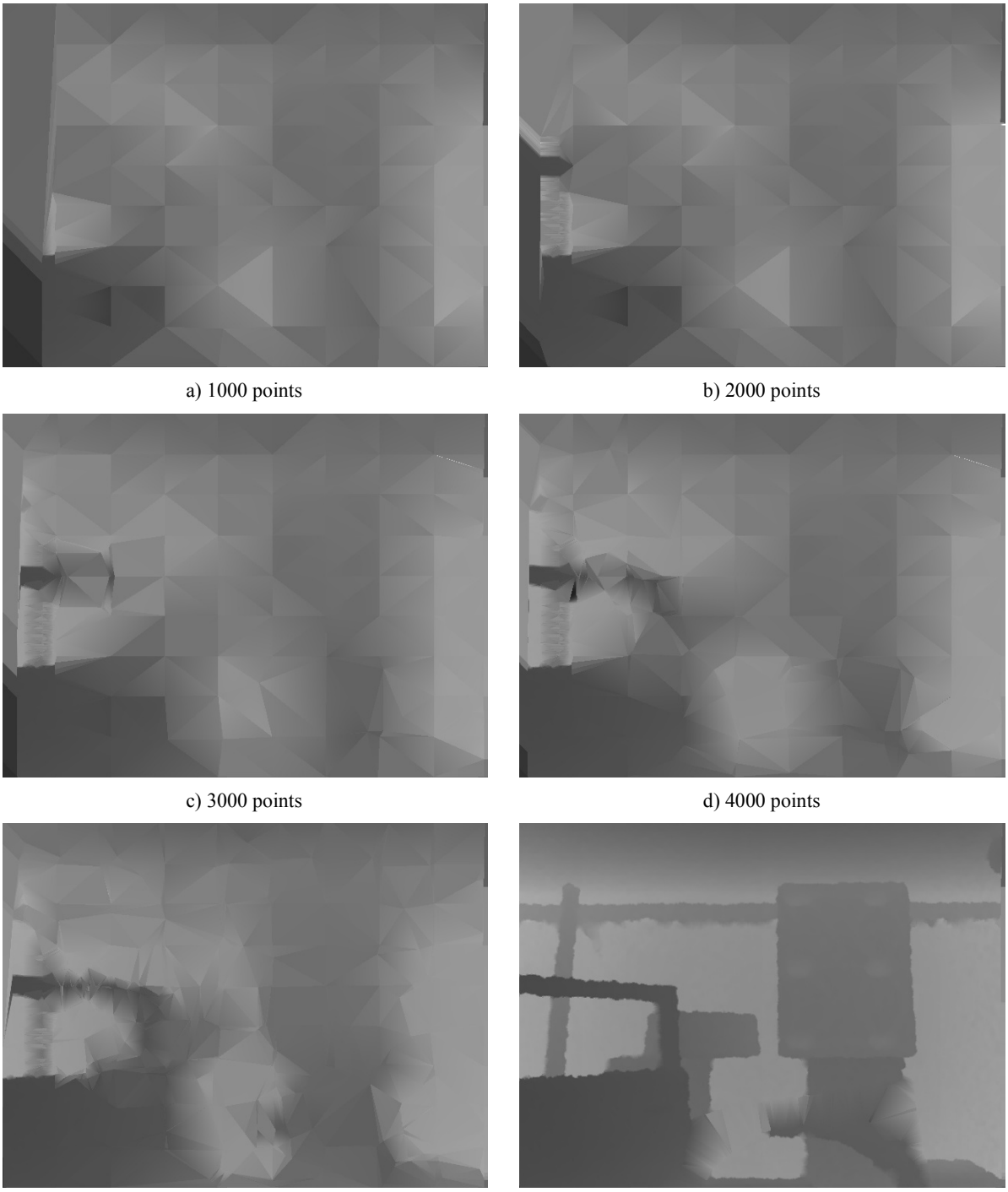


Figure 5.17 - Scene 6: surface mesh based depth map progression after different numbers of points are acquired using the surface mesh improvement based selection process

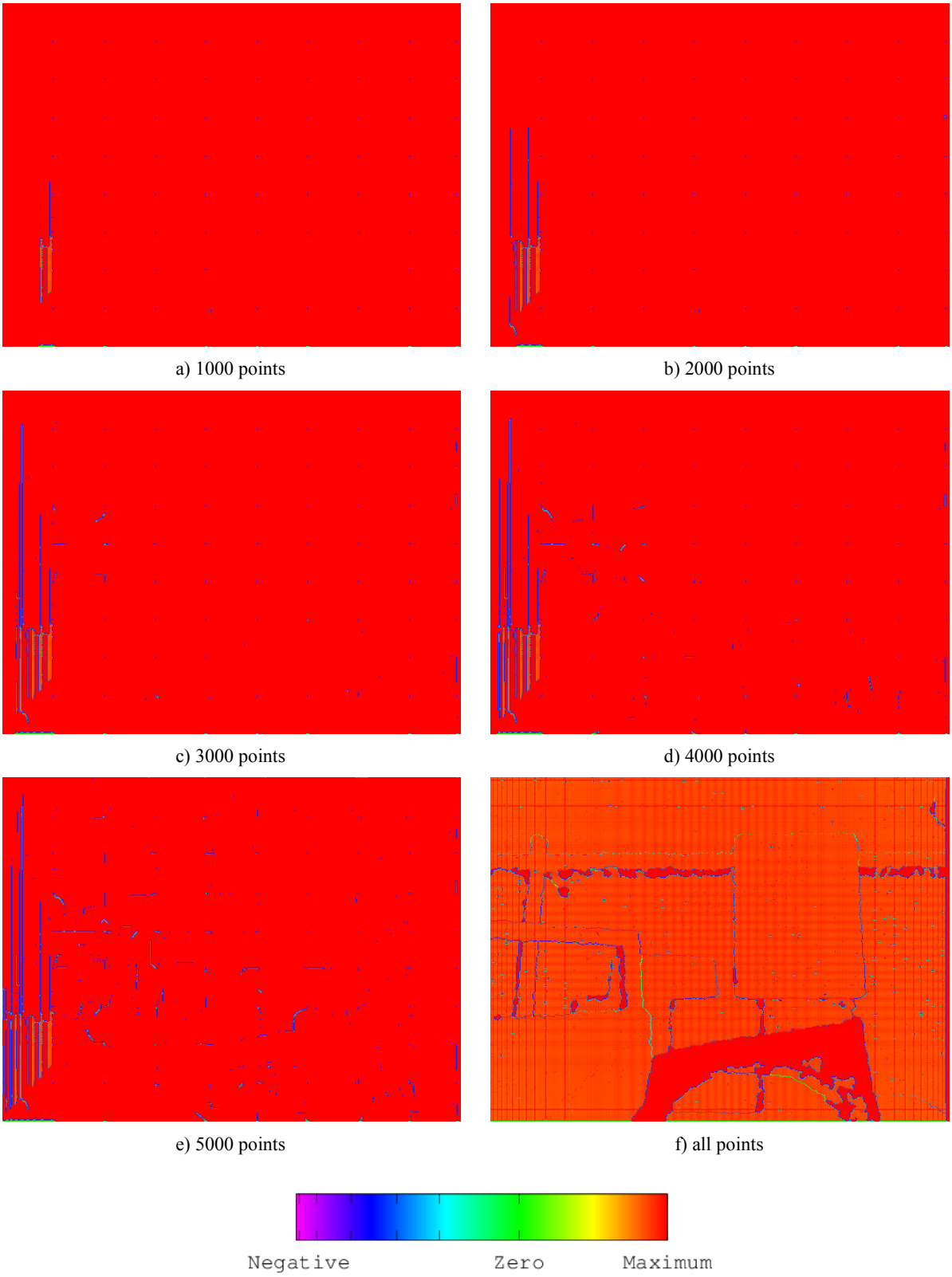


Figure 5.18 - Scene 6: probabilistic occupancy grid improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

5.2.2.4 Discussion on Surface Mesh Improvement Based Selection Process Progression

The surface mesh improvement based selection process performs well at determining the extent of sharp transitions, while balancing out acquisition in regions where there are larger scale features. This proposed method is dependent on the initial sparse scan used to initiate the process, which if it does not sufficiently encompass the scenes as observed by the results for Scenes 2 and 6 presented in Sections D.2.1 and 5.2.2.3 respectively, results in a high improvement region along the edge of the surface mesh improvement map. This is due to the fact that a region outside of the sensor's field of view is unknown to the observer, and hence results in large improvement values that direct the surface mesh improvement based selection process to accurately determine the edges of the scene within the sensor's field of view. The result, is as observed, a zigzag acquisition pattern that starts close to where a transition region is detected, which is close to the edge of the field of view. This zigzag pattern dominates the acquisition until the scene is sufficiently enclosed so that point selection elsewhere in the sensor scanning space can be performed. From Figure D.1 b) and Figure 5.5 b) it is observed that when multiple points are acquired at the same time (N>1), the slow convergence in the mean improvement of the surface mesh improvement measure to the minimal value due to the zigzag pattern is avoided, as points from other triangles are sampled in addition to the zigzag pattern area.

For the other scenarios where the zigzag pattern does not exist, the large scale features become observable after about 3000 points are acquired, and the transition regions become fairly pronounced by 4000-5000 points, which also coincides to the resulting convergence points of the surface mesh mean estimated improvement, and estimated improvement error plots shown in subfigures b) and d) respectively from Figure 5.3 to Figure 5.5 in Section 5.2.1 (and from Figure D.1 to Figure D.3 in Section D.1).

Moreover, the surface mesh improvement map highlights transitions with high improvement, and if taken to the extreme of acquiring all the points, both the surface mesh based and the probabilistic occupancy improvement maps can be used to guide a segmentation, by thresholding for high improvement in the surface mesh based approach, or by thresholding for negative improvement in the probabilistic occupancy grid approach. As can be observed

from the scanning patterns revealed through the progression of the probabilistic occupancy grid improvement map in Figure 5.10, Figure 5.15, and Figure 5.18 (as well as Figure D.6, Figure D.9, and Figure D.12), the surface mesh improvement based selection process tends to select points that are very close to currently sampled locations. This is due to point configurations in the Delaunay triangulation, or as a result of the approximation of the semivariogram made for the surface mesh based improvement calculation (see Section 4.2, Figure 4.11) which results in a slightly higher percentage error when depths very close to existing measure location are estimated. As is observed from the probabilistic occupancy grid improvement measure (Figure 3.10), this behaviour should be mitigable if the probabilistic occupancy grid improvement measure is combined with the surface mesh improvement measure in the combined selection process, which will be examined in Section 5.4.

5.2.3 Execution Performance

The performance of each stage in the decision process algorithm is detailed in this section. The results were obtained during the execution of the decision process algorithm measured while running on an Intel Core I7 2630 QM operating at a Turbo-Boosted nominal frequency of 2.6 GHz. Each of the six scenes used in the analysis in Section 5.2 have two separate tables associated with them, the first for the surface mesh improvement measure computation time requirement, and the second for the probabilistic occupancy grid improvement measure computation time requirement. Both of these timing tables were obtained during the selective acquisition process test cases presented in this section, as driven by the corresponding point selection process. The timing results for Scenes 1, 2, and 4 are presented within this section, while the results for Scenes 3, 5, and 6, are presented in Section D.3. The columns represented in the tables are as follows:

- *Point Acquisition Method*, which is the methodology behind the selection of points, SM DP(N) represents the surface mesh improvement measure based selection process with N points, where $N = \{1,5,10,20\}$, and Random (Mean) represents the mean over 10 runs for random sampling.
- *Number of Points* is the number of points processed for the acquisition method. There may be variations for each dataset due to the binning process used to map

uniformly sampled Cartesian data from the Kinect, to the uniform grid in angular projection space, as well as the possibility of a value not being used due to an estimated zero improvement value.

- *SM and POG Insertion Times* are the accumulated times it took to insert all the points processed into the surface mesh (effectively the time for the Delaunay triangulation algorithm) and the probabilistic occupancy grid respectively.
- *Mean SM and POG Insertion Times Per Point* are the times, on average, it took for a single point to be processed into the surface mesh and the probabilistic occupancy grid respectively.
- *SM and POG Improvement Calculation Times* are the accumulated times it took to calculate the surface mesh improvement measure and the probabilistic occupancy grid improvement measure respectively.
- *SM and POG Total Times* are the total accumulated times that the surface mesh improvement algorithm and the probabilistic occupancy grid improvement algorithm, respectively, took to execute. This is the sum of Insertion Time and Improvement Calculation Time.
- *Percent Time Spent for SM or POG Improvement Calculation over Total SM or POG Time* is defined as: $\frac{Improvement\ Calculation\ Time}{Total\ Time} \times 100\%$.
- *Mean SM and POG Improvement Calculation Time Per Point* are the times, on average, that the surface mesh improvement and the probabilistic occupancy grid improvement calculations, respectively, took per point, and is defined as: $\frac{Improvement\ Calculation\ Time}{Number\ of\ Points}$.
- *Point Selection Process Time* is the accumulated time that the point selection process took to decide the next point to acquire, after the improvement maps are computed.
- *Mean Point Selection Time Per Point* is the mean time per point that the selection process took to decide the next point to acquire and is defined as: $\frac{Point\ Selection\ Process\ Time}{Number\ of\ Points}$.

Note that the acquisition time of the sensor is not taken into account in this data, as is discussed in Section 5.1, because the complete scans, which were fully captured and then

stored in advance, are used to mimic a random access sensor that has identical characteristics to the one used to acquire the data. Furthermore the methods that are presented in this thesis in Chapter 3 are designed to be independent of sensor acquisition technologies, which are characterized by different parameters, including speed of acquisition, and is not a factor directly considered in this thesis.

Table 5.1 presents the effective processing overhead that the surface mesh improvement based selection processes incur. These are calculated based on the surface mesh improvement calculation time as presented in Table 5.2, Table 5.4, Table D.1, Table 5.6, Table D.3, and Table D.5, by calculating the difference in time when the acquisition is guided using the surface mesh improvement based selection processes and the random point selection process. The overhead is between 12 and 40 seconds for the calculation after the insertion of all points for the scenes (1, 3, 4, and 5) where the zigzag scanning pattern is not present, and between 10 and 89 seconds where the zigzag pattern is present (Scenes 2 and 6). Additionally, the addition of more points per iteration decreases this overhead, as there would be less overall manipulations of the involved data structures when more data is processed at once.

	Scene 1 Overhead (sec)	**Scene 2 Overhead (sec)**	**Scene 3 Overhead (sec)**	**Scene 4 Overhead (sec)**	**Scene 5 Overhead (sec)**	**Scene 6 Overhead (sec)**
SM DP(1)	35.617	63.275	36.837	40.088	36.935	88.875
SM DP(5)	26.501	14.309	22.029	23.161	20.34	23.757
SM DP(10)	20.245	12.767	13.329	20.122	18.996	16.558
SM DP(20)	16.86	10.098	12.406	16.049	16.761	14.197

Table 5.1 - Overheads in the calculation of the surface mesh improvement compared to random point selection

The actual amount of time required to make a decision using the surface mesh improvement is low relative to the overall calculation of the improvement measure. Furthermore, as can be observed in the tables of results, as more points are added per iteration, not only is the effective overhead in the calculations reduced, the SM insertion time, SM total time, mean SM improvement calculation time per point, and the point selection process time decrease. Again, this is due to the reduction of the number of manipulation that must occur in the respective data structures used in the surface mesh model and its related improvement measure. The surface mesh improvement measure spends about 85%-90% of the time at

calculating the improvement measure in all cases. In terms of time execution, random point selection outperforms the surface mesh improvement based selection process, due to mainly the overhead added in the calculation time to keep track of the improvement values separately for each triangle, which is reduced as more points are selected at each iteration. Therefore improvement needs to be made to reduce this overhead when designing the selection process that combines both improvement measures, which will be partially achieved due to the additional focus on coverage when compared against the surface mesh improvement based selection process.

Table 5.3, Table 5.5, Table D.2, Table 5.7, Table D.4, and Table D.6 display the timing results for the probabilistic occupancy grid improvement measure to be computed that were determined simultaneously with the surface mesh improvement timing results. As can be observed from the tables, the execution times stay relatively constant regardless of the point selection mechanism used, with slight savings in time appearing for the multipoint selection entries, likely due to less overhead or more effective parallelization of the algorithm. The time required to perform the probability occupancy grid improvement calculations is only 4.5%-7% of the total time for the probability occupancy based algorithm, which is only about 5-9 μs per point and as a result does not lead to large overhead in the process.

In summary, point insertion into the surface mesh model is faster than the probabilistic occupancy grid model (26-65 μs vs. 95-140 μs per point), but as the relative time to calculate the surface mesh is greater than the probabilistic occupancy grid (85%-90% vs. 4.5%-7%), the time it takes for the surface mesh improvement measure is greater than that required for the probabilistic occupancy grid measure (2-17 μs vs. 5-9 μs per point). The variation in the timing results for the surface mesh improvement measure is greater than that of the probabilistic occupancy grid measure. This is due to the fact that the effective number of calculations that must be performed is static regardless of the region that the point is inserted for the probabilistic occupancy grid improvement measure, as the number of calculations is directly related to the truncated region of support, as detailed in Section 3.2. On the other hand, the number of calculations that must be performed in the surface mesh based improvement measure, is directly proportional to the triangle area (in particular the number of points contained within each triangle), as detailed in Section 3.1.2, which means that as a

particular acquisition process favours coverage, overall triangle size becomes smaller, hence the number of calculations are reduced, thereby speeding up the process. This implies that when the two proposed improvement methods are combined, the additional emphasis on coverage should yield reduced computational times, and better performance.

Point Acquisition Method	Number of Points	SM Insertion Time (sec)	Mean SM Insertion Time Per Point (sec)	SM Improvement Calculation Time (sec)	SM Total Time (sec)	Percent Time Spent for SM Improvement Calculation over Total SM Time (%)	Mean SM Improvement Calculation Time Per Point (sec)	Point Selection Process Time (sec)	Mean Point Selection Time Per Point (sec)
SM DP(1)	230477	12.609	5.47E-05	83.751	96.36	86.915	3.63E-04	4.123	1.79E-05
SM DP(5)	230707	8.523	3.69E-05	74.635	83.159	89.751	3.24E-04	1.444	6.26E-06
SM DP(10)	230739	7.442	3.23E-05	68.379	75.82	90.185	2.96E-04	0.989	4.29E-06
SM DP(20)	230751	6.747	2.93E-05	64.994	71.742	90.595	2.82E-04	0.73	3.16E-06
Random (Mean)	254060	6.790	2.67E-05	48.134	54.924	87.638	1.89E-04	0.545	2.15E-06

Table 5.2 - Scene 1: timing data for surface mesh improvement measure

Point Acquisition Method	Number of Points	POG Insertion Time (sec)	Mean POG Insertion Time Per Point (sec)	POG Improvement Calculation Time (sec)	POG Total Time (sec)	Percent Time Spent for POG Improvement Calculation over Total POG Time (%)	Mean POG Improvement Calculation Time Per Point (sec)
SM DP(1)	230477	32.358	1.40E-04	1.966	34.324	5.728	8.53E-06
SM DP(5)	230707	29.628	1.28E-04	1.491	31.119	4.792	6.46E-06
SM DP(10)	230739	29.318	1.27E-04	1.268	30.586	4.146	5.50E-06
SM DP(20)	230751	28.722	1.24E-04	1.305	30.027	4.346	5.66E-06
Random (Mean)	254060	33.356	1.31E-04	1.925	35.281	5.456	7.58e-06

Table 5.3 - Scene 1: timing data for probabilistic occupancy grid improvement measure

Point Acquisition Method	Number of Points	SM Insertion Time (sec)	Mean SM Insertion Time Per Point (sec)	SM Improvement Calculation Time (sec)	SM Total Time (sec)	Percent Time Spent for SM Improvement Calculation over Total SM Time (%)	Mean SM Improvement Calculation Time Per Point (sec)	Point Selection Process Time (sec)	Mean Point Selection Time Per Point (sec)
SM DP(1)	281280	15.797	5.62E-05	117.77	133.57	88.173	4.19E-04	4.295	1.53E-05
SM DP(5)	281480	9.014	3.20E-05	68.804	77.818	88.417	2.44E-04	1.489	5.29E-06
SM DP(10)	281512	7.76	2.76E-05	67.262	75.023	89.656	2.39E-04	1.07	3.80E-06
SM DP(20)	281636	7.253	2.58E-05	64.593	71.845	89.905	2.29E-04	0.827	2.94E-06
Random (Mean)	307200	8.499	2.77E-05	54.495	62.994	86.508	1.77E-04	0.661	2.15E-06

Table 5.4 - Scene 2: timing data for surface mesh improvement measure

Point Acquisition Method	Number of Points	POG Insertion Time (sec)	Mean POG Insertion Time Per Point (sec)	POG Improvement Calculation Time (sec)	POG Total Time (sec)	Percent Time Spent for POG Improvement Calculation over Total POG Time (%)	Mean POG Improvement Calculation Time Per Point (sec)
SM DP(1)	281280	37.163	1.32E-04	2.405	39.568	6.079	8.55E-06
SM DP(5)	281480	34.422	1.22E-04	1.714	36.136	4.743	6.09E-06
SM DP(10)	281512	33.577	1.19E-04	1.642	35.219	4.662	5.83E-06
SM DP(20)	281636	33.665	1.20E-04	1.626	35.291	4.608	5.77E-06
Random (Mean)	307200	38.570	1.26E-04	2.458	41.028	5.99	8.00E-06

Table 5.5 - Scene 2: timing data for probabilistic occupancy grid improvement measure

Point Acquisition Method	Number of Points	SM Insertion Time (sec)	Mean SM Insertion Time Per Point (sec)	SM Improvement Calculation Time (sec)	SM Total Time (sec)	Percent Time Spent for SM Improvement Calculation over Total SM Time (%)	Mean SM Improvement Calculation Time Per Point (sec)	Point Selection Process Time (sec)	Mean Point Selection Time Per Point (sec)
SM DP(1)	253492	13.758	5.43E-05	91.539	105.30	86.935	0.000361	4.23	1.67E-05
SM DP(5)	253695	8.748	3.45E-05	74.612	83.361	89.506	0.000294	1.453	5.73E-06
SM DP(10)	253746	7.85	3.09E-05	71.573	79.424	90.116	0.000282	1.022	4.03E-06
SM DP(20)	253714	7.218	2.84E-05	67.5	74.718	90.339	0.000266	0.809	3.19E-06
Random (Mean)	279299	7.945	2.84E-05	51.451	59.397	86.624	0.000184	0.589	2.11E-06

Table 5.6 - Scene 4: timing data for surface mesh improvement measure

Point Acquisition Method	Number of Points	POG Insertion Time (sec)	Mean POG Insertion Time Per Point (sec)	POG Improvement Calculation Time (sec)	POG Total Time (sec)	Percent Time Spent for POG Improvement Calculation over Total POG Time (%)	Mean POG Improvement Calculation Time Per Point (sec)
SM DP(1)	253492	33.876	1.34E-04	2.193	36.069	6.079	8.65E-06
SM DP(5)	253695	30.691	1.21E-04	1.577	32.268	4.887	6.22E-06
SM DP(10)	253746	30.237	1.19E-04	1.452	31.689	4.582	5.72E-06
SM DP(20)	253714	29.879	1.18E-04	1.48	31.359	4.72	5.83E-06
Random (Mean)	279299	34.840	1.25E-04	2.223	37.062	5.997	7.96E-06

Table 5.7 - Scene 4: timing data for probabilistic occupancy grid improvement measure

5.2.4 Comparison Against Neural Gas Method

This subsection provides a comparison of the proposed surface mesh improvement measure against the neural gas method presented in [6,7], which also appears in [101,102]. As the neural gas method, which is summarized in Section 2.1.1, provides as an output the regions that should be acquired with higher priority. The comparison will be made solely based on the processing of the algorithms after an initial uniform subsampling of various densities from the complete scene datasets, with no further directed evolutions of the respective models considered. The comparison is made across 3 scenes (the computer scene described in Section C.11, the exercise ball scene described in Section C.7, and the fire hose scene described in Section C.20) using 2 different sensors (the Kinect described in Section 4.1.1 and the ASLS described in Section 4.1.2). The various densities of the initial uniform subsampling being used for the comparison are 32x32, 64x64, and 128x128. The surface mesh improvement map based results were executed on an Intel I7-2630 operating at 2.6 GHz, coded in C++, while the results for the neural gas method were obtained from code running on a separate machine running Matlab.

As the ASLS data contains many erroneous outliers, these were removed by hand for this subsection to ensure a clear comparison of the performance of each method. Furthermore, to prevent regions that have no sampled locations from dominating the surface mesh improvement map, these regions were removed and replaced with black pixels (which represent no improvement) for visualization purposes. The surface mesh improvement map has been scaled in each case to be between zero (black) and one (white), with darker shades of grey representing a lesser estimated potential improvement compared to lighter shades. For the neural gas method the mapping is inverted, blacker regions representing areas that should be reacquired with higher density, and white representing sections that have a lower priority.

In Scene 11 of the computer as acquired using the Kinect, both methods determine that the boundary edges of the objects (monitor, keyboard, and chassis) should be acquired with higher priority (Figure 5.19). The results displayed in Figure 5.20, for Scene 7 of the exercise ball acquired using the Kinect, indicate via the surface mesh improvement map that the out edge of the ball, and the base used to prop it up, are the regions of priority to acquire

(in white). Meanwhile the neural gas method highlights not only these edges, but also several locations within the ball (in black). The surface mesh improvement method, in Scene 20 of the fire hose station acquired using the Kinect, indicates, as shown in Figure 5.21, that the edges around the black pipe, the yellow pipe, the red enclosure, and the fire extinguisher should be acquired with higher density, with few larger improvement values (light gray) in the region where the hose is bundled up. On the other hand, the neural gas method, while highlighting these edges, indicates that the hose area is of greater importance to acquire in higher density through the higher density of black coloured pixels (Figure 5.21). Furthermore, as the sampling density increases, the neural gas method focuses the attention of future acquisitions in this hose region, while the surface mesh improvement map focuses more on the depth transitions in the scene. These three sets of results illustrate that both methods perform similarly over planar surfaces, but on curved and textured surfaces, the surface mesh improvement method focuses more on edges of objects, while the neural gas method focuses more on increasing coverage on these curved and texture surfaces.

The results of the same scenes, but acquired using the ASLS, which are not only noisier, but are also sparser, indicate similar results as obtained in the case where the scenes were acquired with the Kinect in regards to planar surfaces, as put into evidence in Figure 5.22 and Figure 5.24. However, in Scene 7 of the exercise ball when acquired with the ASLS, the surface mesh improvement map indicates that most of the ball should be acquired in higher detail, while the neural gas method indicates that only the interior of the ball should be acquired in higher resolution (Figure 5.23). This behaviour is likely due to the larger concentration of outliers about the ball still present in the scene. As the sampling density increases from 32x32 to 128x128, the region that should be acquired in more detail increases in size, according the neural gas method, while the surface mesh improvement measure only highlights a few regions, scattered throughout the scene. This demonstrates that the surface mesh improvement measure is more resilient to outliers and sampling density variations than the neural gas method.

Examining the results obtained using both methods indicates that as the initial sampling density increases from 32x32 to 128x128, the regions that require additional sampling become sharper, independently of the overall resolution of the sensor used to acquire the

data. Furthermore, as can be seen by comparing the scenes acquired with the Kinect (Figure 5.19 - Figure 5.21) against that of the ASLS (Figure 5.22 - Figure 5.24), both methods perform well even when the overall acquisition has plenty of gaps, with the surface mesh improvement measure performing better in the presence of noise. The surface mesh improvement map tends to provide higher improvement values (lighter shades) along edges and contours, as these are areas where there is distinct difference between regions, while the neural gas highlights regions (darker shades) that tend to be more uniformly spread, and hence has better coverage. Of final note, the neural gas technique requires an initial subsampling be performed on the scene, but the proposed surface mesh improvement measure requires no initial samples, providing it greater flexibility in usage.

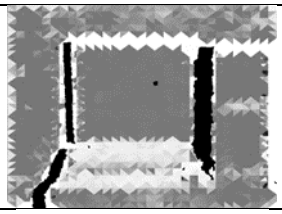
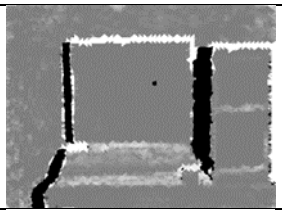
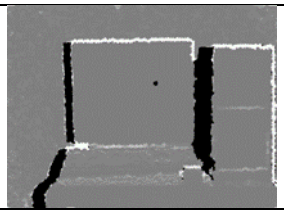
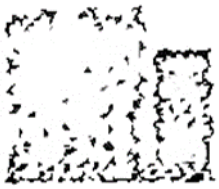
Initial uniform sampling density	32x32	64x64	128x128
Surface mesh improvement measure			
Selected regions from the neural gas method			

Figure 5.19 - Region selection comparing the surface mesh improvement measure against the neural gas method when applied to the computer scene acquired with the Kinect.

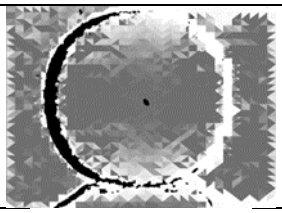
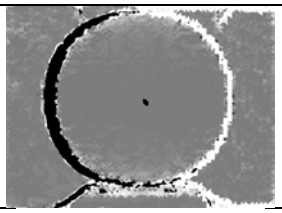
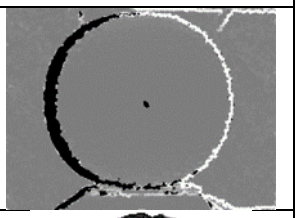
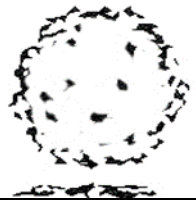
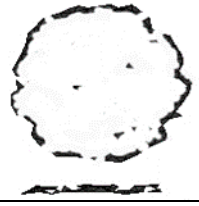
Initial uniform sampling density	32x32	64x64	128x128
Surface mesh improvement measure			
Selected regions from the neural gas method			

Figure 5.20 - Region selection comparing the surface mesh improvement measure against the neural gas method when applied to the exercise ball scene acquired with the Kinect.

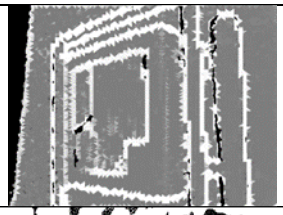
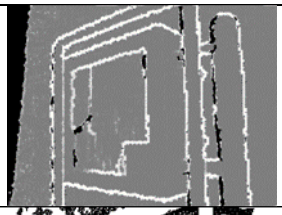
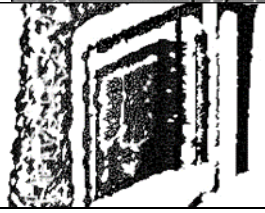
Initial uniform sampling density	32x32	64x64	128x128
Surface mesh improvement measure			
Selected regions from the neural gas method			

Figure 5.21 - Region selection comparing the surface mesh improvement measure against the neural gas method when applied to the fire hose scene acquired with the Kinect.

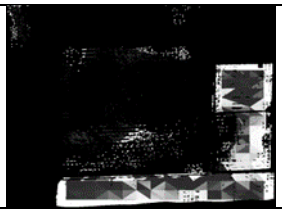
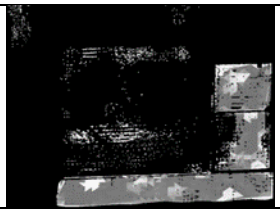
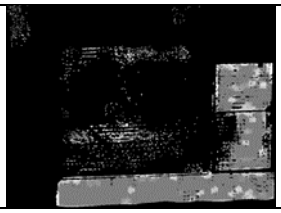
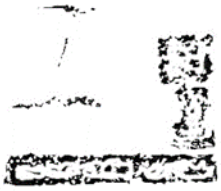
Initial uniform sampling density	32x32	64x64	128x128
Surface mesh improvement measure			
Selected regions from the neural gas method			

Figure 5.22 - Region selection comparing the surface mesh improvement measure against the neural gas method when applied to the computer scene acquired with the ASLS.

Initial uniform sampling density	32x32	64x64	128x128
Surface mesh improvement measure			
Selected regions from the neural gas method			

Figure 5.23 - Region selection comparing results surface mesh improvement measure against the neural gas method when applied to the exercise ball scene acquired with the ASLS.

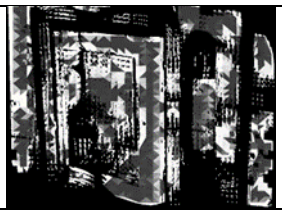
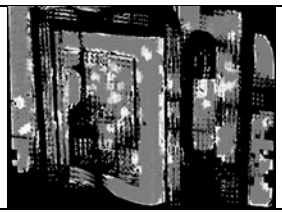
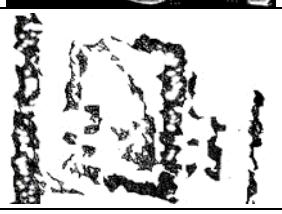
Initial uniform sampling density	32x32	64x64	128x128
Surface mesh improvement measure			
Selected regions from the neural gas method			

Figure 5.24 - Region selection comparing the surface mesh improvement measure against the neural gas method when applied to the fire hose scene acquired with the ASLS.

Table 5.8 and Table 5.9 indicate the execution performance of each method for each set of results. While a formal direct comparison of timing information is not valid as computation for both methods were performed on different architectures and computers, the overall trends in performance can be compared. From these tables, it can be observed that the neural gas method execution time is consistent for each initial sampling density, with the 32x32 density taking between 8 and 10 seconds to execute, the 64x64 density taking between 32 and 38 seconds to execute, and the 128x128 density taking between 140 and 160 seconds to execute. This indicates that the neural gas method's execution time is primarily driven by the initial sampling density, regardless of scene, or sensor resolution. The surface mesh improvement measure on the other hand is dependent on both the initial sampling density and on the sensor resolution, as this is used as the resolution of the improvement map.

	Computer scene		Exercise ball scene		Fire hose scene	
	Improvement map	Neural gas	Improvement map	Neural gas	Improvement map	Neural gas
32x32	0.68 s	9.5 s	0.66 s	9.6 s	0.66 s	9.3 s
64x64	0.82 s	37.2 s	0.83 s	37.5 s	0.85 s	36.5 s
128x128	1.39 s	153.0 s	1.41 s	153.9 s	1.41 s	150.0 s

Table 5.8 - Algorithm execution time of each method under various initial sampling densities for each scene acquired with the Kinect

	Computer scene		Exercise ball scene		Fire hose scene	
	Improvement map	Neural gas	Improvement map	Neural gas	Improvement map	Neural gas
32x32	13.4 s	9.8 s	16.7 s	8.9 s	19.5 s	8.8 s
64x64	25.5 s	36.8 s	19.8 s	35.8 s	40.0 s	32.3 s
128x128	42.4 s	161.6 s	15.2 s	142.3 s	81.7 s	140.2 s

Table 5.9 - Algorithm execution time of each method under various initial sampling densities for each scene acquired with the ASLS

By observing the timing information between each initial sampling density found in Table 5.8 and Table 5.9, and then calculating the ratios between them, the relative execution time scaling between the various sampling densities can be computed. This scaling is presented in Table 5.10 for the scenes acquired with the Kinect and Table 5.11 for the scenes acquired with the ASLS. From these tables, it can be observed that the neural gas method scales near linearly with an increase of samples, as an increase by a factor of four in the initial sampling density that occurs when proceeding from 32x32 to 64x64 and from 64x64 to 128x128 results in a factor of about four times increase in execution time, and this hold true regardless of the scene, sensor resolution, or sampling density. The surface mesh improvement measure illustrates a scaling of execution time between initial sampling densities of 32x32 to 64x64 and 64x64 to 128x128, which represents a factor of four times increase in sampling density, of between 1.2 and 2 (except for the case of the exercise ball acquired with the ASLS). This means that when the sampling density is quadrupled, the execution time for the neural gas method is quadrupled, but the execution time for the surface mesh improvement measure is at most doubled. This indicates that surface mesh improvement method out scales the neural gas method, in terms of initial sampling density by a factor of 2. The performance gain seen in the 128x128 result of the exercise ball seen in Table 5.9 is likely due to the fact that the improvement measure is highly sensitive to CPU cache hits and misses, which means that triangles with fewer pixels can be calculated much faster than triangles with more pixels, as all the data can fit into the cache.

	Computer scene		Exercise ball scene		Fire hose scene	
	Improvement map	Neural gas	Improvement map	Neural gas	Improvement map	Neural gas
32x32 to 64x64	1.21 x	3.92 x	1.26 x	3.91 x	1.29 x	3.92 x
32x32 to 128x128	2.04 x	16.11 x	2.14 x	16.03 x	2.14 x	16.13 x
64x64 to 128x128	1.70 x	4.11 x	1.70 x	4.10 x	1.66 x	4.11 x

Table 5.10 - Detailing the scaling of execution time between various initial sampling densities for each scene acquired with the Kinect

	Computer scene		Exercise ball scene		Fire hose scene	
	Improvement map	Neural gas	Improvement map	Neural gas	Improvement map	Neural gas
32x32 to 64x64	1.90 x	3.76 x	1.19 x	4.02 x	2.05 x	3.67 x
32x32 to 128x128	3.16 x	16.49 x	0.91 x	15.99 x	4.19 x	15.93 x
64x64 to 128x128	1.66 x	4.39 x	0.77 x	3.97 x	2.04 x	4.34 x

Table 5.11 - Detailing the scaling of execution time between various initial sampling densities for each scene acquired with the ASLS

5.2.5 Discussion on the Surface Mesh Improvement Based Selection Process

The assumption made earlier in this thesis, namely that the number of points to provide a good accurate representation of the scene is much less than a complete dense scan, is quantitatively confirmed in Section 5.2.1 and visually verified in Section 5.2.2. The analyses revealed that most large scale features can be put into evidence before 5000 points are acquired in most scenes. Comparing this result to the naïve dense acquisition method for the Kinect sensor that produces up to 307200 points, this represents a possible 60 times savings in number of acquired points, which can prove very beneficial when performing acquisitions using a much slower but more accurate range sensors than the Kinect device.

Section 5.2.1, which provides the quantitative analysis of the proposed surface mesh improvement based selection process indicates that it tends to outperform the random point selection process in terms of the surface mesh improvement metrics, implying better directed acquisition towards scene accuracy. The probabilistic occupancy grid improvement metrics, on the other hand, showed that the random point selection process performed better than the proposed surface mesh improvement based selection process at guiding the acquisition to provide coverage of the scene. Section 5.2.2, further highlighted the fact that the surface mesh improvement based selection process focuses more on acquiring the accurate locations of features in the scene as the underlying surface mesh improvement map had higher levels of improvement for sharp transition regions.

Section 5.2.2 also exposed some of the weaknesses of the surface mesh improvement based selection process, which includes the fact that points tend to be chosen next to each other along the edges of objects, and includes the case when the initial acquisition does not encompass the complete region of the scene (i.e. no points along the edges) that results in significantly larger improvement values in these regions. This causes a zigzag acquisition pattern, as observed in Figure D.6 and Figure 5.18. The effects of the zigzag pattern on the

progression of the scanning of the scene can be mitigated by the acquisition of multiple points at a time, as illustrated by the mean improvement results in Figure D.1 a),b), and Figure 5.5 a),b). It was also observed that the acquisition of a larger number of points (greater than 5) at a time can lead to higher mean improvement levels. These weaknesses could be reduced by combining the surface mesh improvement measure with a complementary technique, such as the probabilistic occupancy grid improvement measure presented in Section 3.2, and whose corresponding selection process performance will be examined Section 5.3, which follows.

The time based performance of the selection process using the surface mesh improvement measure, as discussed in Section 5.2.3, adds significant overhead to the calculation of the coinciding improvement calculations, mainly due to the tracking of the improvement values for each triangle, which reduces in effective cost when acquiring many points at the same time. The probabilistic occupancy grid improvement measure takes less time to process than the surface mesh improvement measure. An analysis detailing the effective performance of the selection processes in regards to execution time and other metrics of performance is performed later, in Section 5.4, which analyses the performance of the selection processes against the proposed combined selection process.

Finally, as seen in Section 5.2.4 the surface mesh improvement measure is seen to outperform the neural gas method for low resolution sensors, such as the Kinect, and has the potential to outperform the neural gas sensor for high resolution if enough points are initially sampled. Furthermore, the execution time of the surface mesh improvement measure scales better in terms of number of points inserted than does the neural gas method by a factor of two.

5.3 Probabilistic Occupancy Grid Improvement Based Selection Process Evaluation

The evaluation of the proposed probabilistic occupancy grid improvement based selection process, which was developed in Section 3.3.2 is presented in this section. The evaluation of this selection process is performed in regards to a comparison with the random point selection process, over the same 6 scenes that were used in Section 5.2 when performing the evaluation of the surface mesh improvement based selection process. However, only the

results of one run of the random point selection process is used in the comparison instead of the mean of 10 runs as was evaluated previously in Section 5.2, and as a result the plots will present different characteristics.

The first subsection compares the performance of the proposed probabilistic grid improvement based selection process against that of the random point selection process, using the same quantitative metrics employed in Section 5.2.1. The second subsection provides a qualitative analysis illustrating the progression of the probabilistic occupancy grid improvement based selection process for the 6 selected scenes. The third subsection provides a qualitative analysis of the performance of the proposed probabilistic occupancy grid improvement measure as a 3D point cloud segmenter.

5.3.1 Quantitative Analysis

The metrics being employed for the evaluation of the proposed probabilistic occupancy grid improvement based selection process in this section are the same metrics that were used in Section 5.2.1, namely the surface mesh mean estimated improvement, surface mesh estimated improvement error, the probabilistic occupancy grid mean estimated improvement, and the probabilistic occupancy grid estimated improvement error. The surface mesh improvement measures, as driven by these two selection processes, are used as metrics here as they provide indications as to how each selection process performs at accurately representing a scene, while the probabilistic occupancy grid improvement measures provide indications as to how each selection process performs at driving acquisition towards enhancing coverage.

Graphs are presented which show the progression of each metric, based upon the locations selected by the probabilistic occupancy grid improvement based selection process and by the random point selection process. In all cases, the vertical range of the plots has been adjusted such that the main characteristics are better observed. For each set of results, the plots are labelled with the following convention:

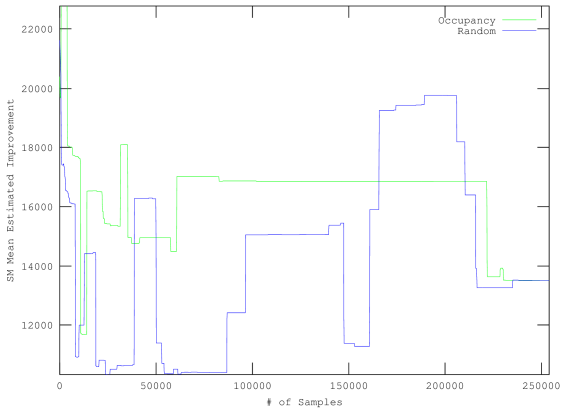
a) Surface mesh mean estimated improvement over all points.
b) Surface mesh estimated improvement error over all points.
c) Probabilistic occupancy grid mean estimated improvement over all points.

d) Probabilistic occupancy grid estimated improvement error over all points.

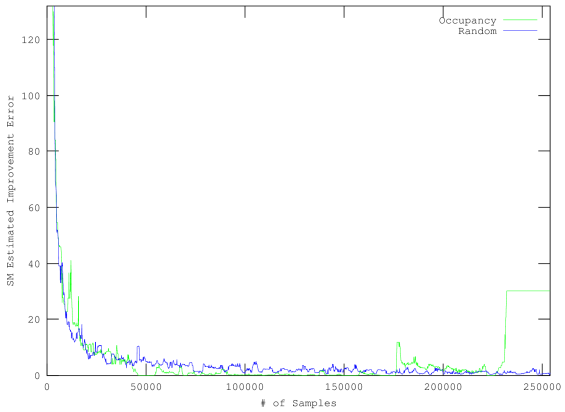
The legend on the plots referring to *Occupancy*, represents the graph of the corresponding metric as points are acquired using the probabilistic occupancy grid improvement based selection process, while legend referring to *Random* represents the graph of the corresponding metric as points are acquired using the using the random point selection process over a single run. This section contains the analysis for Scenes 1, 3, and 6, while the analysis for Scenes 2, 4, and 5 are presented in Section D.4.

5.3.1.1 Analysis of Scene 1

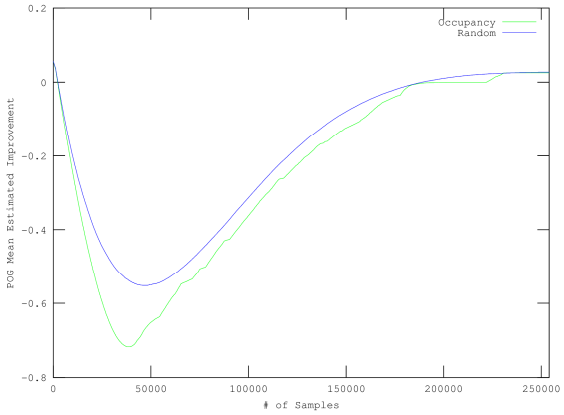
Scene 1 (see Section C.1) is the first scene to evaluate the algorithms against. Figure 5.25 a), showing the surface mesh mean estimated improvement, illustrates chaotic noisy behaviour for both selection processes, likely due to the large region on the right hand side of the field of view that is unacquirable by the Kinect, as the depth surpasses the maximum depth that the Kinect is able to perceive. The plot of the surface mesh estimated improvement error, Figure 5.25 b), on the other hand shows that the random point selection process converges to the minimum error a little earlier than that of the probabilistic occupancy grid improvement based selection process, which indicates that the random point selection process is slightly better at acquiring the scene accurately with fewer points. Figure 5.25 c), a plot of the probabilistic occupancy grid mean estimated improvement, shows that the probabilistic occupancy grid improvement based selection process provides better overall coverage of the scene when compared to the random point selection process, with the minimum occurring after only about 40000 points being inserted. Figure 5.25 d) showing the probabilistic occupancy grid estimated improvement error, also indicates that the probabilistic occupancy grid improvement based selection process performs better at scene coverage than the random point selection process, as its curve is lower for most of the acquisition cycle.



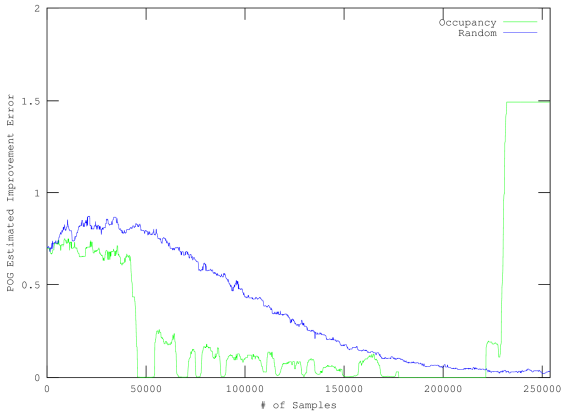
a) Surface mesh mean estimated improvement over all points



b) Surface mesh estimated improvement error over all points



c) Probabilistic occupancy grid mean estimated improvement over all points

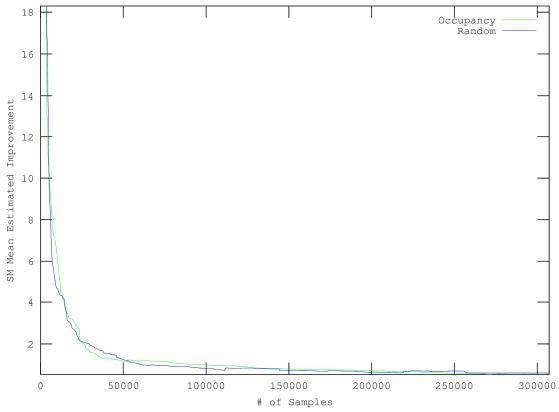


d) Probabilistic occupancy grid estimated improvement error over all points

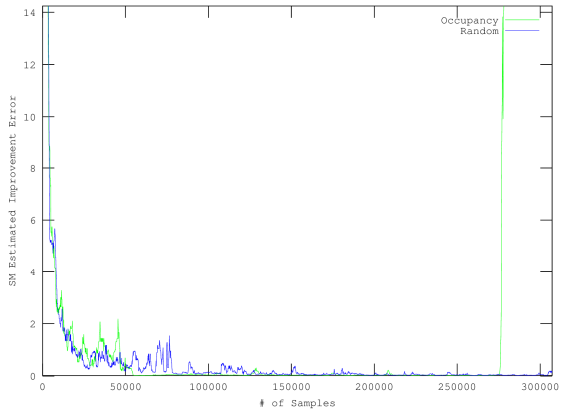
Figure 5.25 - Scene 1: mean improvement and improvement error trend graphs

5.3.1.2 Analysis of Scene 3

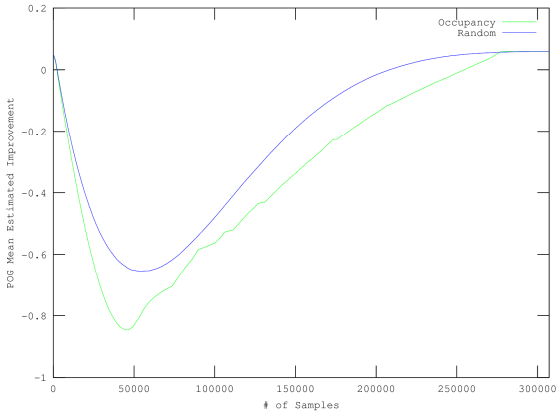
Scene 3 (see Section C.3) is the scene in this section to evaluate the algorithms against. Figure 5.26 a), which contains the plot of the surface mesh mean estimated improvement metric, shows that the curve representing the random point selection process converges slightly earlier to the minimum value, indicating that it is slightly better at accurately capturing the scene earlier when compared to the probabilistic occupancy grid improvement based selection process. The plot of the surface mesh estimated improvement error, Figure 5.26 b), shows that this error, when guided by the probabilistic occupancy grid improvement based selection process, may indeed converge earlier than that produced using the random point selection process and with less noise. However, it increases dramatically at the end, likely due to the locations along the depth transition being captured last.



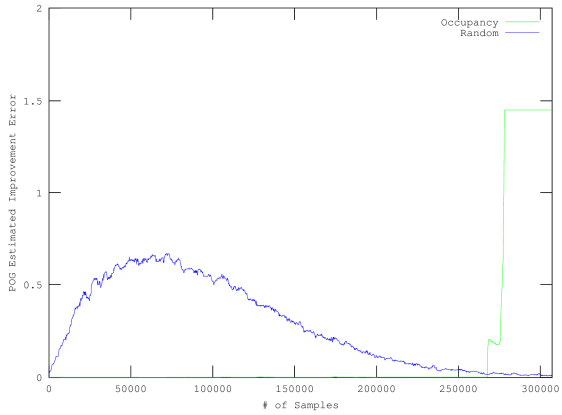
a) Surface mesh mean estimated improvement over all points



b) Surface mesh estimated improvement error over all points



c) Probabilistic occupancy grid mean estimated improvement over all points

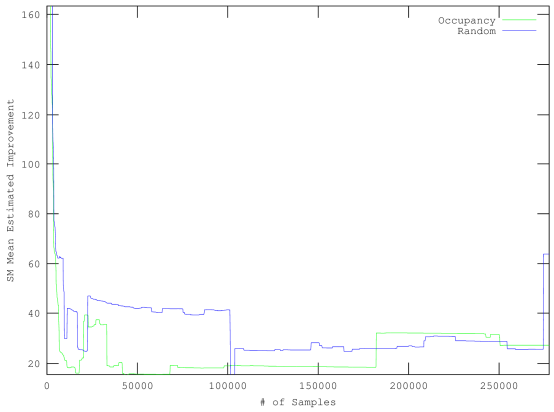


d) Probabilistic occupancy grid estimated improvement error over all points

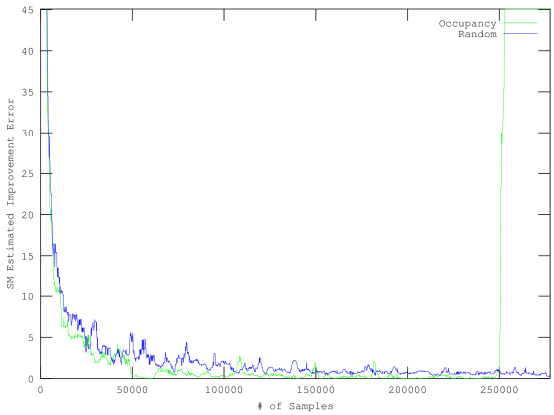
Figure 5.26 - Scene 3: mean improvement and improvement error trend graphs

5.3.1.3 Analysis of Scene 6

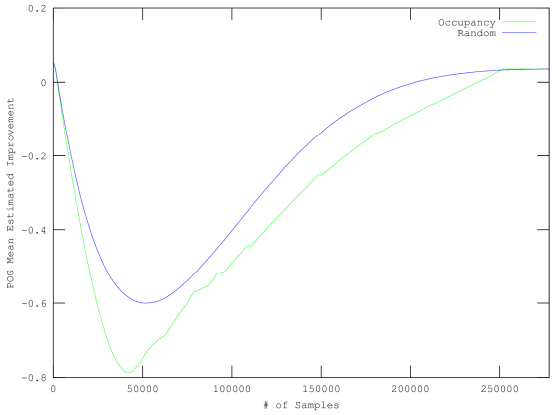
Scene 6, shown in Section C.6, shows a cluttered scene within a laboratory environment. The plot of the surface mesh mean estimated improvement, Figure 5.27 a), and the plot of the surface mesh estimated improvement error, Figure 5.27 b), each indicate that both selection processes perform similarly well at providing scene accuracy. The probabilistic occupancy grid mean estimated improvement plot, Figure 5.27 c), indicates that the probabilistic occupancy grid improvement based selection process performs better at scene coverage, than the random point selection process, with the former process's curve reaching a lower minimum, with only about 40000 points inserted. Figure 5.27 d), reinforces the probabilistic occupancy grid improvement based selection processes ability at providing better scene coverage, as the corresponding curve is near zero until near the end, while the random selection process has a much higher value for the majority of the plot.



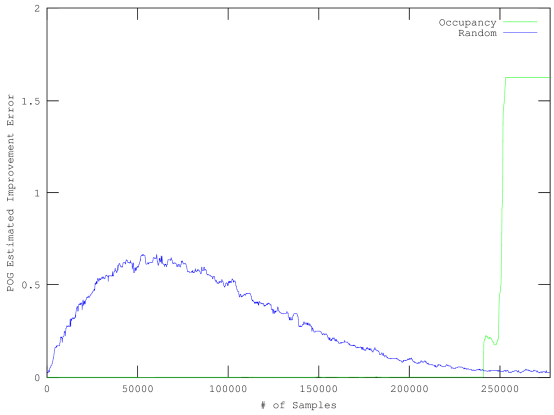
a) Surface mesh mean estimated improvement over all points



b) Surface mesh estimated improvement error over all points



c) Probabilistic occupancy grid mean estimated improvement over all points



d) Probabilistic occupancy grid estimated improvement error over all points

Figure 5.27 - Scene 6: mean improvement and improvement error trend graphs

5.3.1.4 Summary of Quantitative Analysis

All six scenes studied (Scenes 1, 3, and 6 in this section and Scenes 2, 4, and 5 in Section D.4) exhibit similar overall trends for Figure 5.25 - Figure 5.27 (and for Figure D.13 - Figure D.15). For the subfigures a), which is the graph of surface mesh mean estimated improvement containing all the points available from the respective point clouds, neither of the selection processes compared in this section perform consistently better than the other at this metric, and although they both eventually converge to a value, both tend to be fairly noisy curve. This indicates that neither of the selection processes is better for driving acquisition for more accurate representation of the scene, which is the expected outcome as the progressive occupancy grid improvement selection process focusses on improving coverage of the scene, and the random selection process favours scene coverage at it chooses points based on a uniform random distribution. For the subfigures b), which is the graph of surface mesh estimated improvement error containing all the points available from the

respective point clouds, also, similar to the previous case, indicates that neither of the selection processes perform better than the other for this metric. Additionally, as observed by the consistent large spike in this error for the curves produced by the probabilistic occupancy grid improvement based selection process, this method likely chooses points on edges and depth transitions closer to the end. This reinforces the fact that the probabilistic occupancy grid improvement based selection process performs poorly at capturing scene accuracy.

The probabilistic occupancy grid improvement based metrics, as are represented in subfigures c) and d), show without a doubt that the probabilistic occupancy grid improvement based selection process does indeed provide better coverage of the scene as compared to the random point selection process. Subfigures c) illustrate this behaviour by the quite similar graphs that show the probabilistic occupancy grid improvement selection reaches a minimal value of probabilistic occupancy grid mean estimated improvement that is significantly lower and with fewer points captured (between 40000-45000 points) than the random point selection process. Furthermore, subfigures d) illustrate that the probabilistic occupancy grid improvement based selection process produces close to ideal behaviour for the probabilistic occupancy grid estimated improvement error metrics, by staying close to zero error, until the end, where it increases drastically. While the random point selection process does a very good job at providing scene coverage, it still does perform worse than the former method.

In summary, visualized using the surface mesh improvement metrics, both the probabilistic occupancy grid improvement based selection process and the random point selection process do a poor job at focusing on scene accuracy. As for scene coverage, the random point selection process does well at this task, but the probabilistic occupancy grid improvement based selection process does a much better task, as put into evidence by the plots of the probabilistic occupancy grid improvement metrics.

5.3.2 Selection Process Progression

This section will show the progression of the probabilistic occupancy grid improvement based selection process, as it acquired points for each of the six scenes. This will be performed qualitatively by showing the progression of the point locations chosen by the

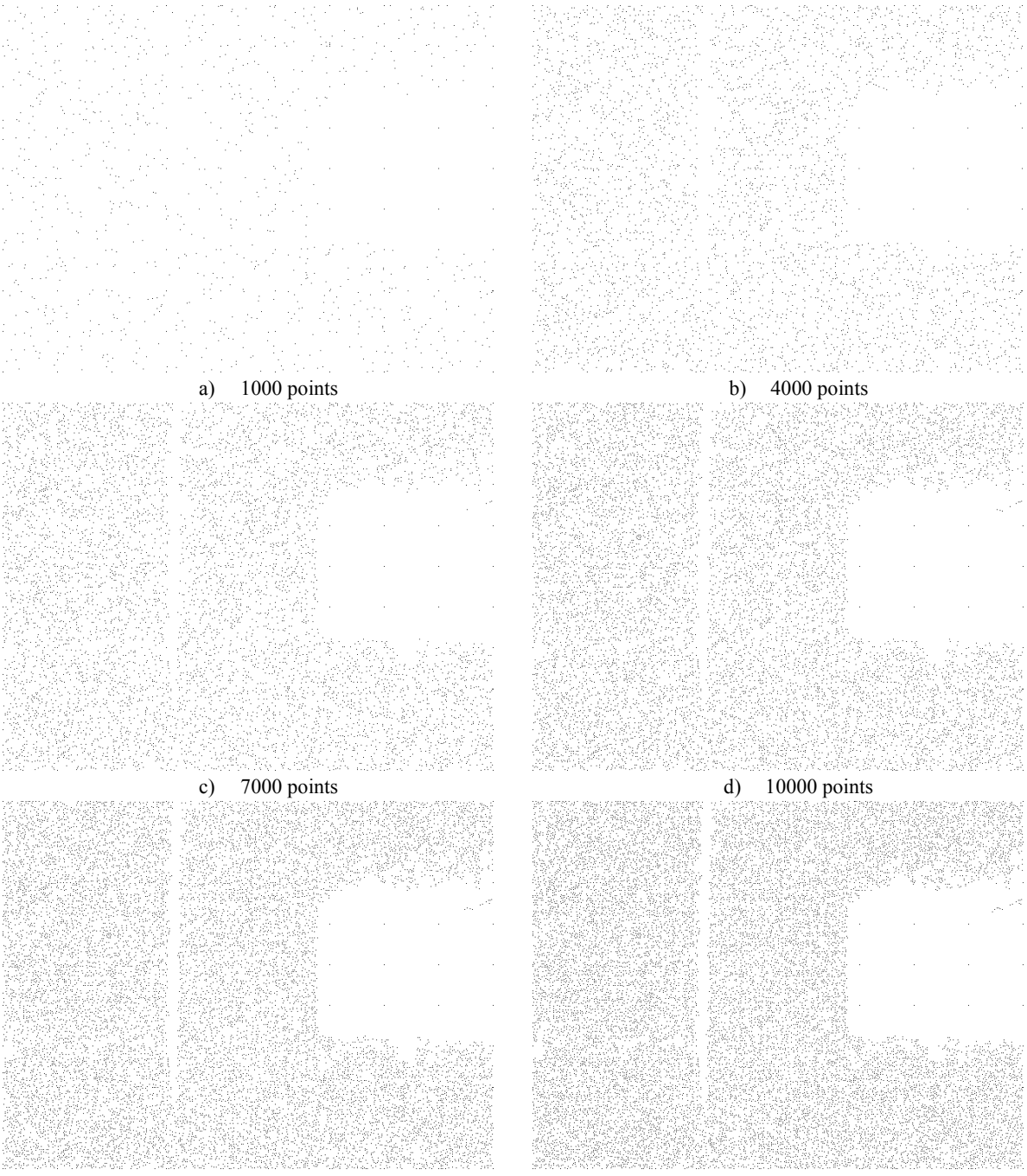
probabilistic occupancy grid improvement based selection process, and the evolution of the probabilistic occupancy grid model itself, by plotting the corresponding voxels in 3D. In the chosen point location maps, each pixel represents a location within the sensor's, Kinect in this case, field of view. A black pixel represents a locations that have been acquired, and white pixels correspond to pixels that are either not yet acquired or unacquirable (with the exception of unacquirable points contained in the initial subsampling, which are also plotted in black for reference). Note that this representation was not used in Section 5.2.2, as the focus was on how the surface mesh improvement model evolved as more points were chosen to be inserted. The evolution will be shown for 1000, 4000, 7000, 10000, 13000, and 16000 points selected and inserted. For the representation of the probabilistic occupancy grid, all voxels that have $P(Occ) > 0.5$ are displayed. Furthermore, for the first scene, a comparison against the random point selection process will be provided. The specific analysis on scenes 2, 4, and 5 are available in Section D.5, as the analysis shares much in common with Scenes 1, 3, and 6.

5.3.2.1 Scene 1: Pillars in a Parking Garage

Figure 5.28 presents the progression of the acquired points that were inserted into the probabilistic occupancy grid model, as selected by the probabilistic occupancy grid based selection process for Scene 1 shown in Section C.1 and acquired using the Kinect. As can be observed from the progression of selected points that were inserted into the model, the point density is uniform within the regions that the Kinect can perceive depths, as opposed to the regions that contain unacquirable points, such as the region to the middle right of the scene, and the region to the immediate left of the pillar . Furthermore, the point density increases more or less uniformly as the progression moves from 1000 points in Figure 5.28 a) to 16000 point in Figure 5.28 f). Figure 5.29, which is the progression of the acquired points that were inserted into the probabilistic occupancy grid model as selected by the random point selection process, illustrates that the random point selection process also acquires points that are distributed uniformly, regardless of the subfigure observed. While the random point selection process does choose points that are distributed uniformly, there are clusters of connected points that can be observed clearly in Figure 5.29 e) and f), which are not present in the progression of the probabilistic occupancy grid improvement based selection process. This qualitatively verifies the result from Section 5.3.1, which is that the probabilistic

occupancy grid improvement based selection process does better at choosing points to enhance coverage, compared to the random point selection process.

Figure 5.30, which is the probabilistic occupancy grid representation corresponding to the point locations chosen by the probabilistic occupancy grid improvement based selection process, shows that not only is space where objects are located, but also empty space (locations that the Kinect can perceive, but which is beyond the 5m range that limits the probabilistic occupancy grid) is chosen equally to be covered by the probabilistic occupancy grid improvement based selection process. As more point are added into the model, from 1000 points in Figure 5.30 a) to 16000 points in Figure 5.30 f), the features become more densely represented by voxels, but their edges are not clearly defined, even after 16000 points are inserted. Figure 5.31 details the progression of the probabilistic occupancy grid model as guided by the random point selection process. The progression from 1000 points in Figure 5.31 a), until 16000 point in Figure 5.31 f), similarly shows that the random point selection process chooses points to acquire uniformly, regardless of whether those location correspond to empty regions of the scene, or if objects are present. Furthermore, with the increasing point acquisition, the objects in the scene become more fully covered by voxels. However, unlike with the probabilistic occupancy grid improvement based selection process, the random point selection process will sometimes choose neighbouring locations that create larger clusters of neighbouring voxels, which can be observed in Figure 5.31 e) and f). This again verifies that the probabilistic occupancy grid improvement based selection process provides better coverage than the random point selection process. Finally, it should be noted that the probabilistic occupancy grid does not perform any large interpolation between voxels implicitly by the base representation. In other words, voxels that do not correspond to empty space, but that are not visible in these images (corresponding to unknown space, where the probability of occupancy is 0.5) are not inherently predicted by the probabilistic occupancy grid representation.

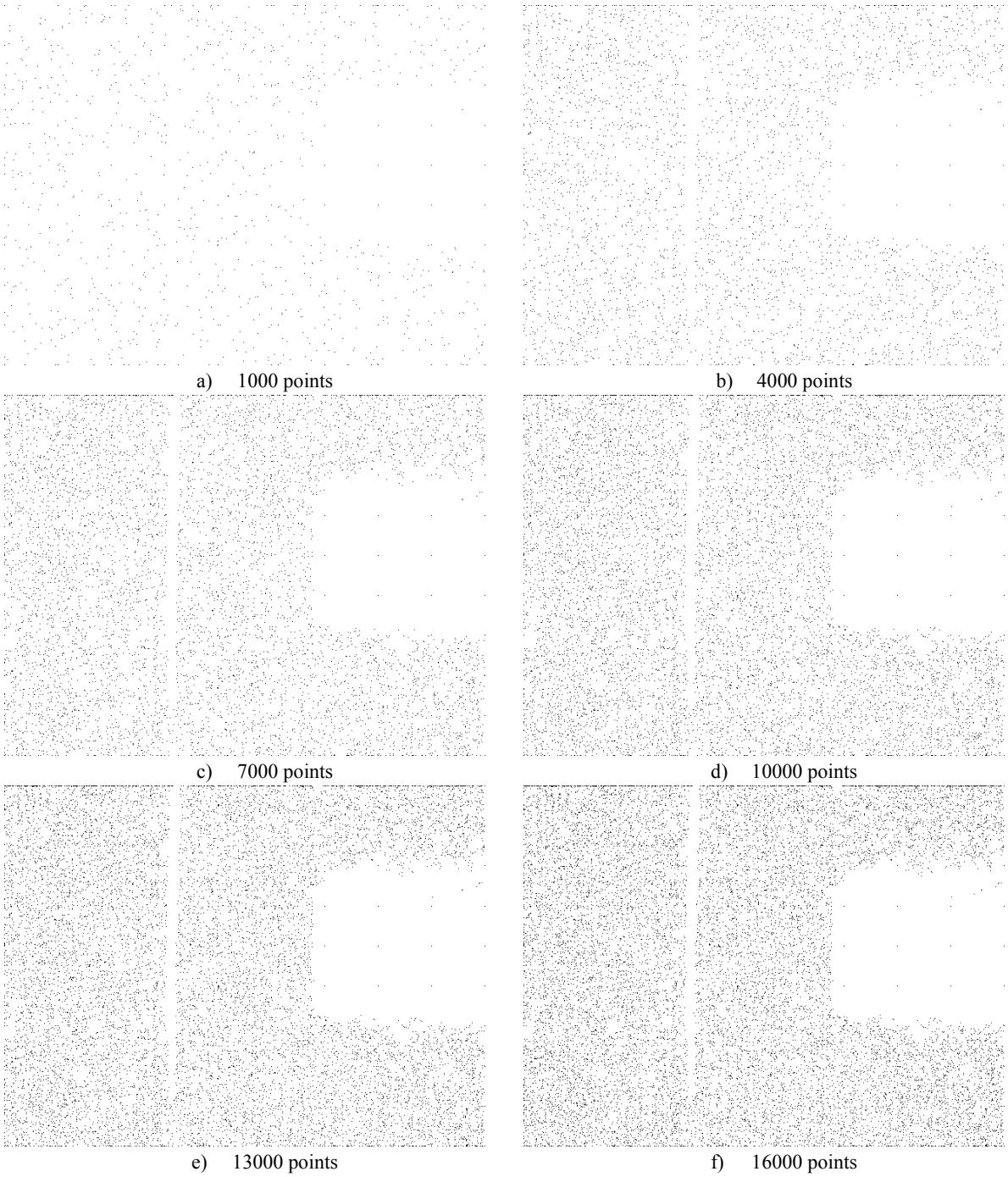


a) 1000 points b) 4000 points

c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

Figure 5.28 - Scene 1: point selection map progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process

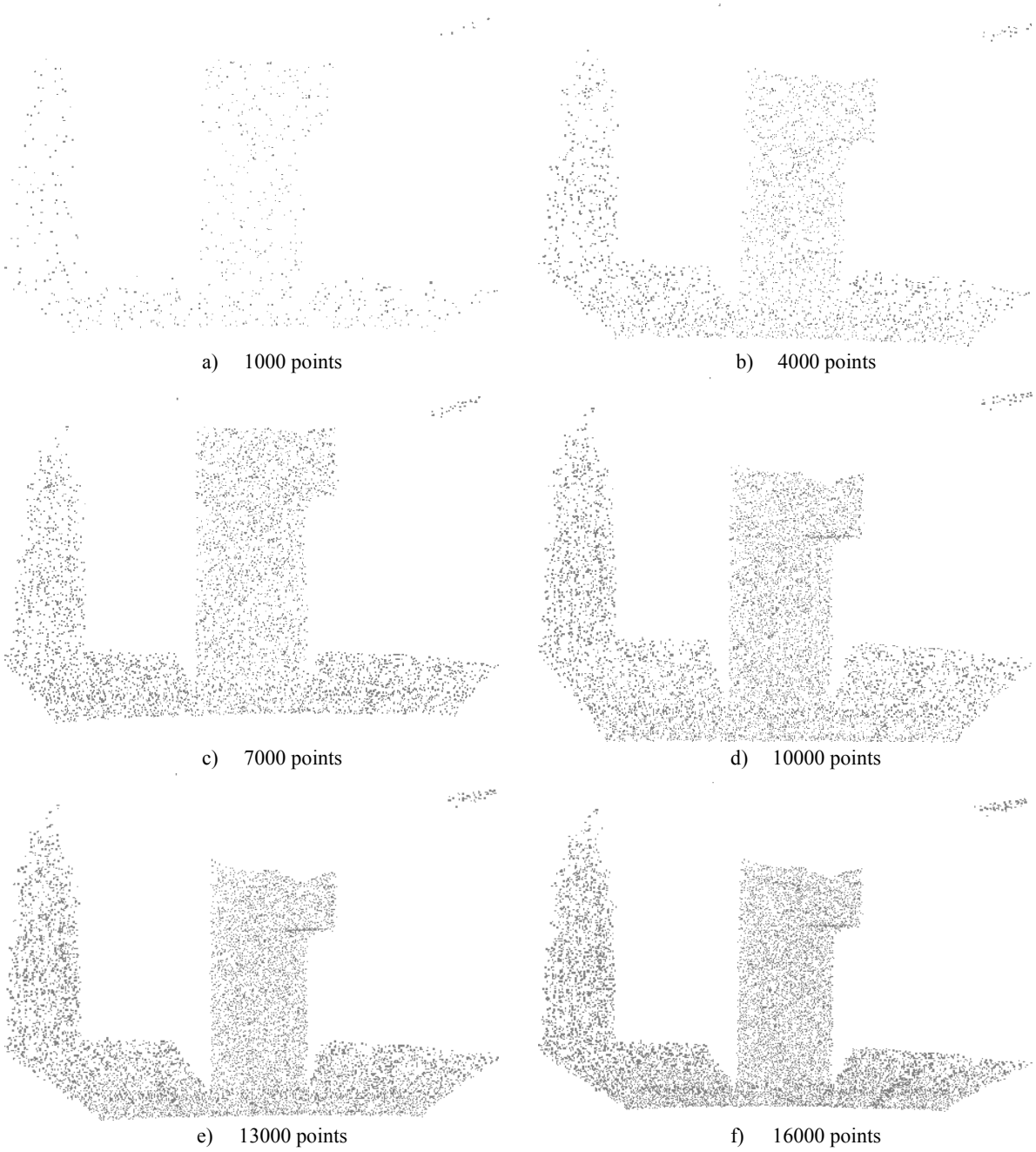


a) 1000 points b) 4000 points

c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

Figure 5.29 - Scene 1: point selection map progression after different numbers of points are acquired using the random point selection process

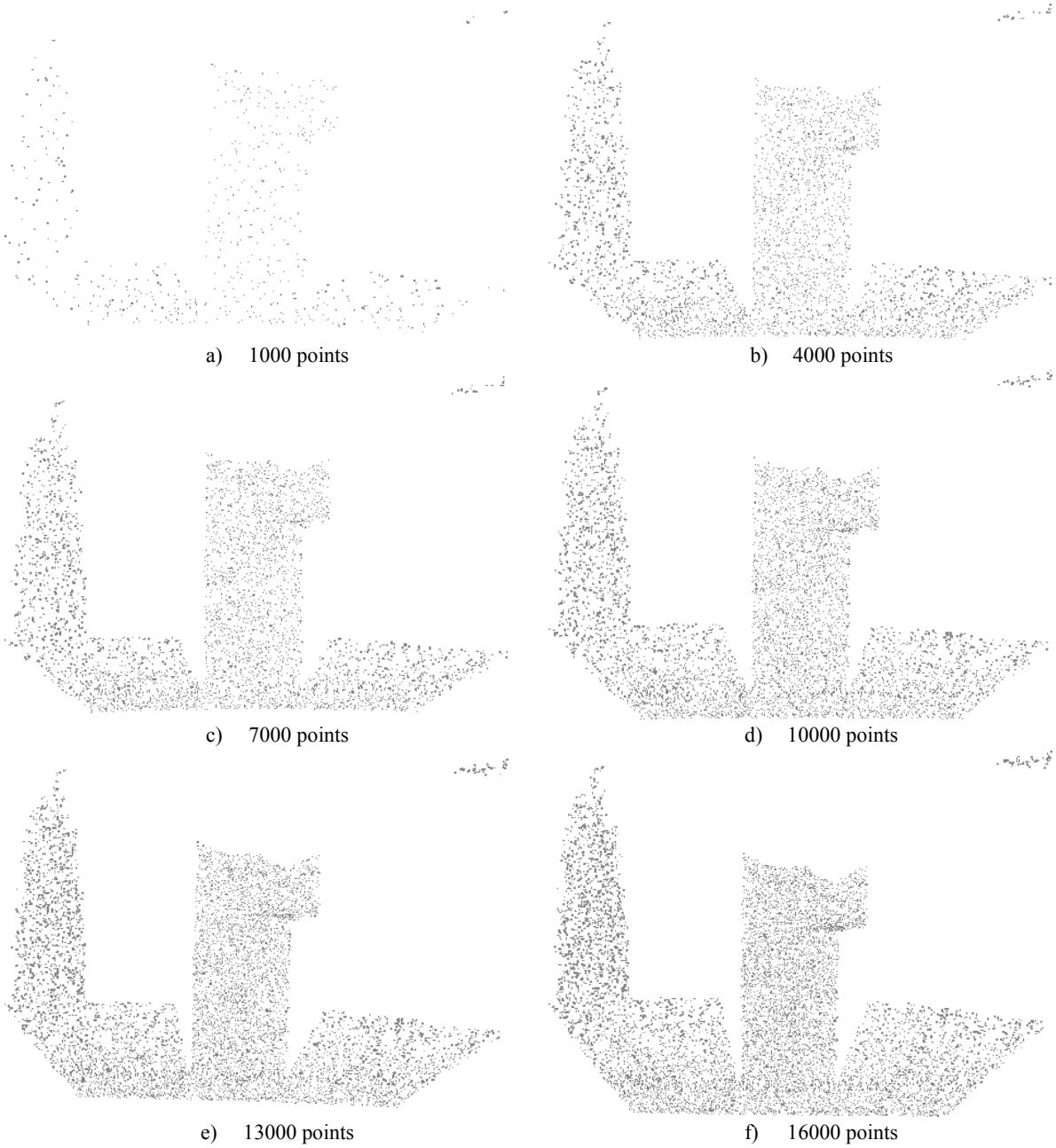


a) 1000 points b) 4000 points

c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

Figure 5.30 - Scene 1: probabilistic occupancy grid model (for P(Occ)>0.5) progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process



a) 1000 points b) 4000 points

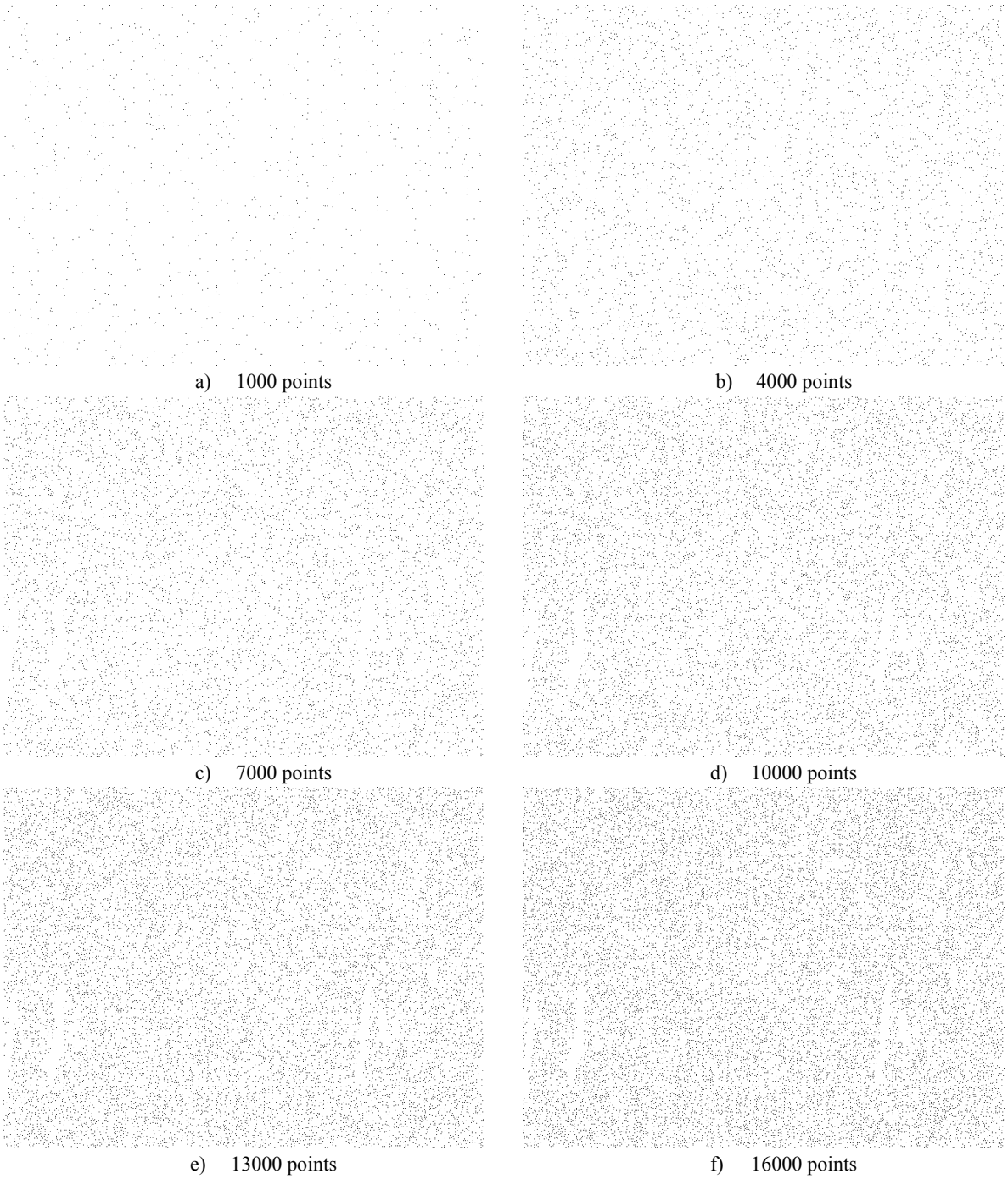
c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

Figure 5.31 - Scene 1: probabilistic occupancy grid (for P(Occ)>0.5) progression after different numbers of points are acquired using the random point selection process

5.3.2.2 Scene 3: Cluttered Corner in a Parking Garage

In Scene 3 of the cluttered parking garage (shown in Section C.3), the progression of the locations acquired in the scene is present in Figure 5.32. As with the previous scenes, the proposed probabilistic occupancy grid improvement based selection process choses point locations that are uniformly distributed, but are not directly adjacent to each other. This is observable regardless of the number of locations selection for acquisition, as shown by Figure 5.32 a-f). This verifies that the proposed probabilistic occupancy grid improvement based selection process does indeed drive the acquisition towards better scene coverage. Observing the relative densities of voxels contained in the probabilistic occupancy grid model for $P(Occ) > 0.5$ presented in Figure 5.33, further verifies this, as the cones and the walls have approximately equivalent voxels densities across their surfaces as the points are progressively and selectively acquired and integrated into the model from 1000 points in Figure 5.33 a) to 16000 point in Figure 5.33 f). Again note, that although the voxels are fairly dense along the surface, there is still much unknown space between them, which is not interpolated or represented accurately.

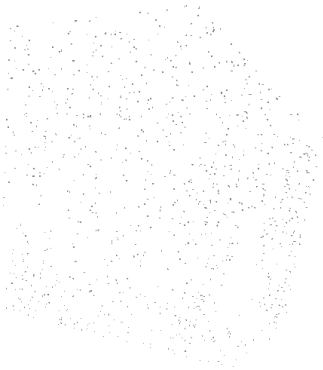


a) 1000 points b) 4000 points

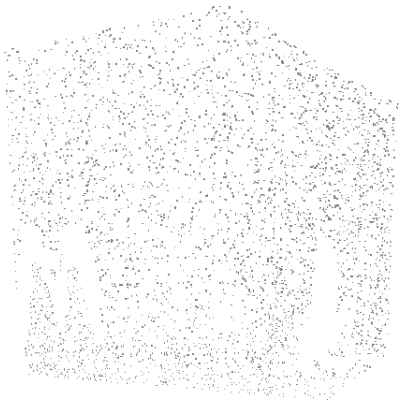
c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

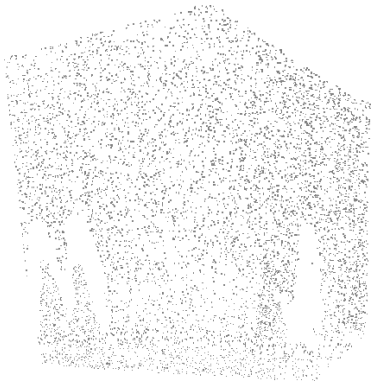
Figure 5.32 - Scene 3: point selection map progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process



a) 1000 points



b) 4000 points



c) 7000 points

d) 10000 points

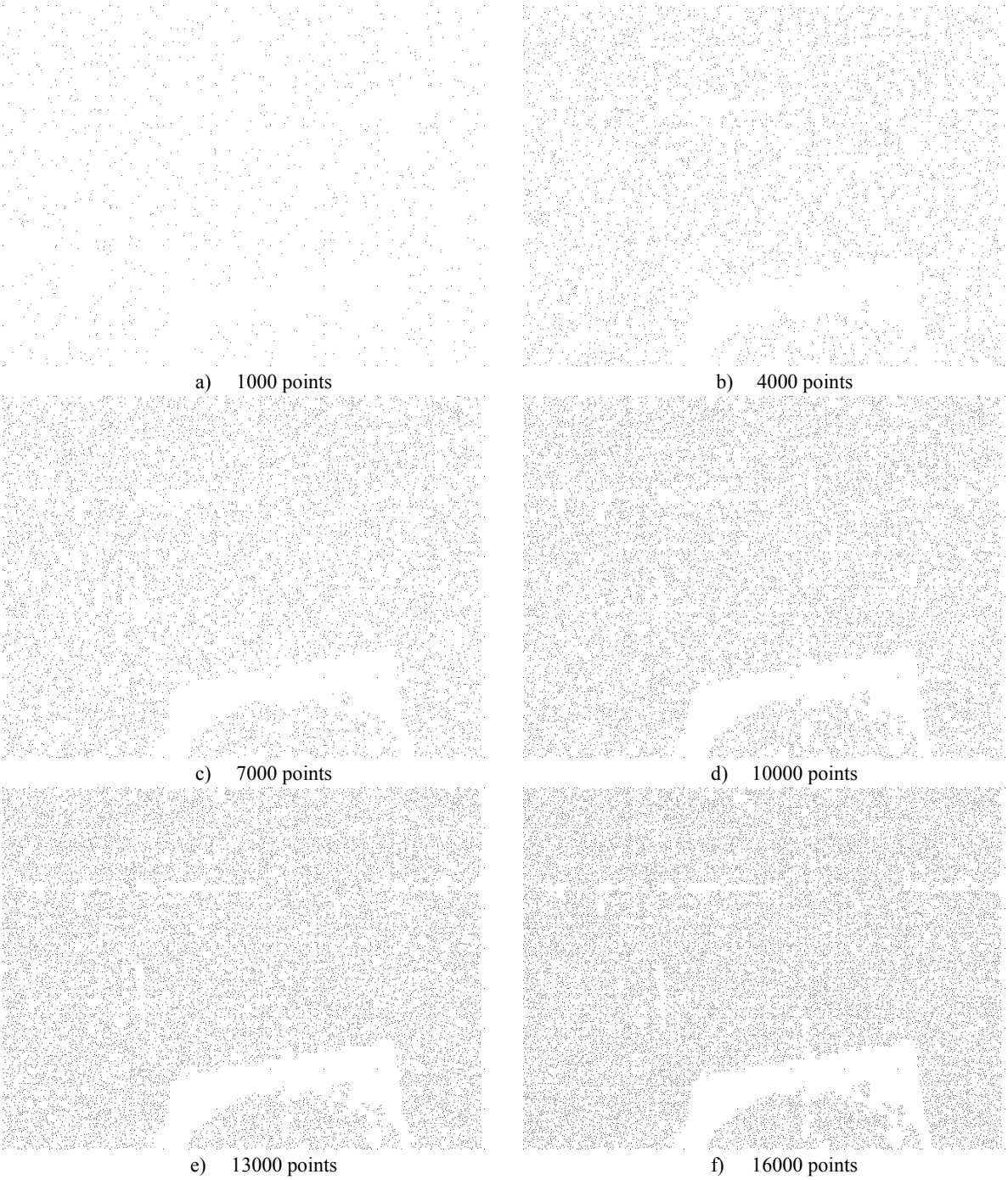
e) 13000 points

f) 16000 points

Figure 5.33 - Scene 3: probabilistic occupancy grid model (for P(Occ)>0.5) progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process

5.3.2.3 Scene 6: Cluttered Scene in a Lab

In this case, for Scene 6 shown in Section C.6 that consists of a collection of cluttered object, the progression of point locations selectively chosen by the probabilistic occupancy grid improvement based selection process is presented in Figure 5.34. The relative point density for surfaces that the Kinect can perceive (most of the objects except for the fender), remains similar across all surfaces as the progression of chosen points increases from 1000 point in Figure 5.34 a) to 16000 in Figure 5.34 f), with no points being acquired immediately adjacent to already captured locations. This verifies the quantitative observation made in Section 5.3.1, that the probabilistic occupancy grid improvement based selection process favours acquiring points to further improve scene coverage. Figure 5.35 of the progression of the probabilistic occupancy grid model for $P(Occ) > 0.5$ shows that the various objects in the scene (mock car door, pallet, cardboard cylinder, and boxes) have similar voxel densities along their surfaces, regardless of the number of points selectively inserted into the model. Note that although the voxel densities present after 16000 points have been selectively inserted are such objects can be recognized, Figure 5.35 f), there are still plenty of unknown voxels along the various object surfaces that are not interpolated intrinsically by the probabilistic occupancy grid representation.

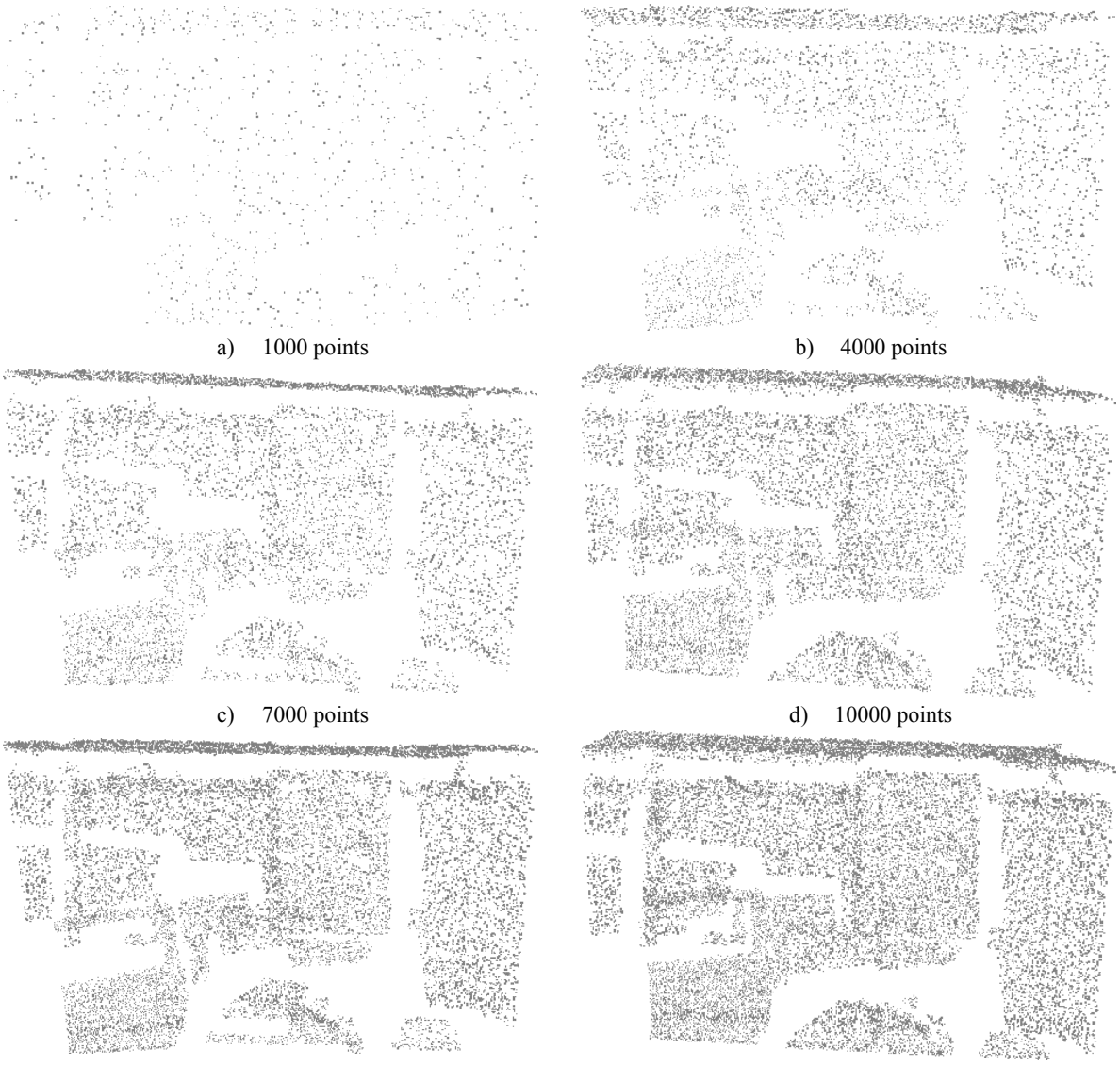


a) 1000 points b) 4000 points

c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

Figure 5.34 - Scene 6: point selection map progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process



a) 1000 points

b) 4000 points

c) 7000 points

d) 10000 points

e) 13000 points

f) 16000 points

Figure 5.35 - Scene 6: probabilistic occupancy grid model (for P(Occ)>0.5) progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process

5.3.2.4 Discussion of Probabilistic Occupancy Grid Improvement Based Selection Process Progression

The proposed probabilistic occupancy grid improvement based selection process, as observed in the point location progression maps (Figure 5.28, Figure 5.32, Figure 5.34, Figure D.16, Figure D.18, and Figure D.20), choses points that are uniformly distributed, but unlike locations chosen by the random point selection process (Figure 5.29), do not contain points that are chosen to be directly adjacent to others. This verifies the quantitative analysis made in Section 5.3.1, that the proposed probabilistic occupancy grid improvement based selection process does indeed direct acquisitions towards better scene coverage than does the random point selection process. An additional observation can be made from these maps. Namely that both the objects, and the empty space (locations that the Kinect can perceive, but are outside of the 5.0m of support used in this thesis) are each chosen to be acquired with similar density. The progression of the selective insertion of points into the probabilistic occupancy grid model (Figure 5.30, Figure 5.33, Figure 5.35, Figure D.17, Figure D.19, and Figure D.21) show that each object in the respective scenes, at each separate stage in the acquisition as illustrated by the progression from 1000 in subfigures a) to 16000 points in subfigures f), are covered by equivalent density of voxels. Furthermore, it can be observed that although the coverage improves, the accurate representation of the edges and transitions in the scene are not clearly defined. Finally it should be noted that as the probabilistic occupancy grid does not implicitly perform large scale interpolation, the space between selected points, and hence the known voxels along the surfaces, remains to correspond to unknown locations.

5.3.3 Exploration of Additional Applications of the Probabilistic Occupancy Grid Improvement Selection Process

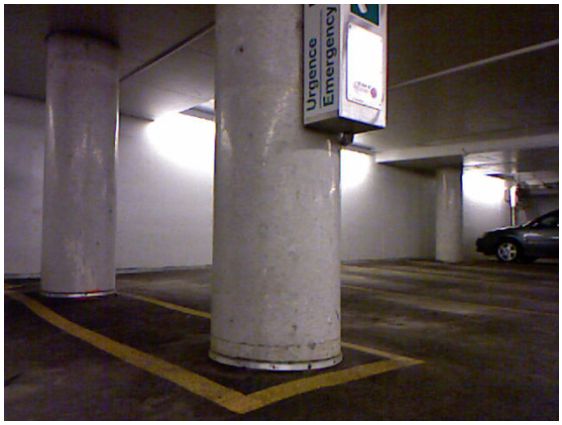
This section details the behaviour of the probabilistic occupancy grid improvement map when all available points contained in the scene have been inserted. The probabilistic occupancy grid improvement based selection process, as described in Section 3.3.2, does not focus on specific features, but it performs well for determining broad regions that need more acquisitions to improve coverage.

In order to demonstrate the possible segmentation application that is exhibited by the probabilistic occupancy grid improvement measure that occurs after all the locations in the scene have been inserted into the model, the selection of both high improvement (${DP'}_2(imp_{POG}, \text{threshold})$ in eq. (3.86)) and negative improvement (${DP''}_2(imp_{POG}, \text{threshold})$ in eq. (3.87)) regions is performed on the improvement map (where threshold in the first case being selected to the highest improvement, in red, and the second case being set to the lowest improvement, that is zero, in blue, present in the map).

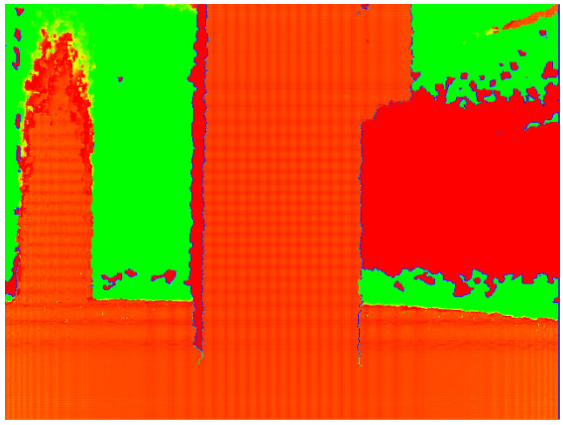
The following subsections present the results of the application of these two selection processes that are based on the probabilistic occupancy grid improvement measure. The results for Scenes 1, 4 and 6 are each presented in a separate subsection below, with the additional results and analysis for Scenes 2, 3 and 5 being presented in Section D.6.

5.3.3.1 Scene 1 - Pillars in a Parking Garage

Figure 5.36 b) shows the resulting probabilistic occupancy grid improvement map for the scene in Figure 5.36 a) of the pillars in a parking garage. The thresholding of high improvement results in Figure 5.36 c), while Figure 5.36 d) shows the thresholding of the negative improvement regions. Note that the highest improvement, Figure 5.36 c), lies in regions where there is occlusion within the Kinect sensor itself (where the IR projector can project but the IR camera cannot see such as the side of the main pillar), where the Kinect sensor cannot get depth readings due to empty space exceeding the range of the sensor (greater than 9.0 m), and where the transitions are formed when the surface gradually extends beyond the far range depth of support of the probabilistic occupancy grid (5.0m). The negative improvement region (Figure 5.36 d)) lies where there is a sharp depth transition between regions, as well as between areas that are fully acquired and those that cannot be acquired due to limitations of the sensor.

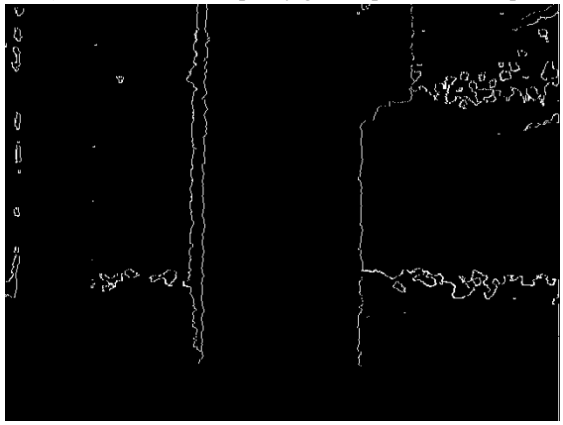


a) RGB image of scene



b) Probabilistic occupancy grid improvement map

c) High improvement regions (DP'_2)

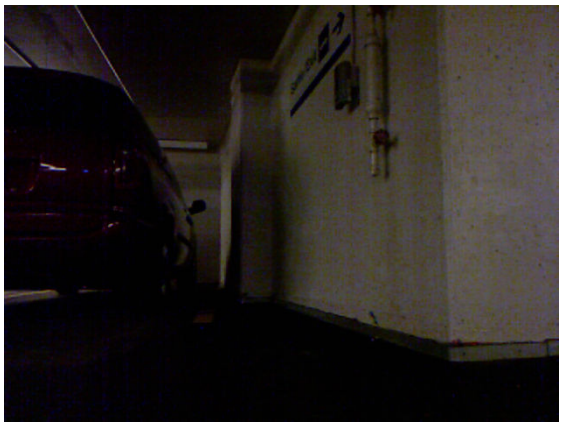


d) Negative improvement regions (DP''_2)

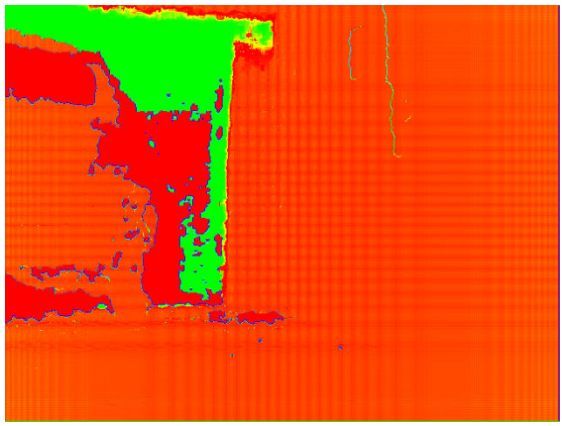
Figure 5.36 - Probabilistic occupancy grid improvement based selection processes on Scene 1

5.3.3.2 Scene 4 - Car and Walls in a Parking Garage

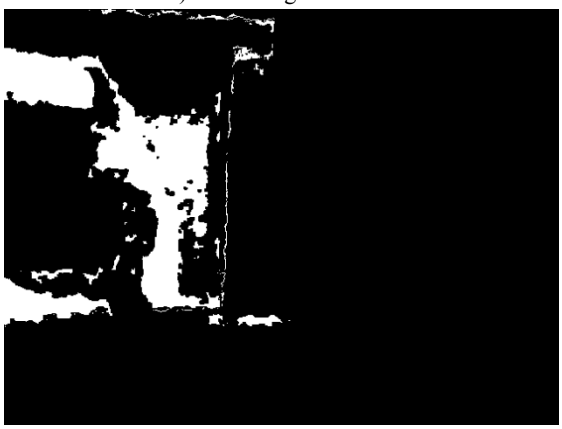
The results observed from Figure 5.37 c) illustrate that location of the maximum improvement from the probabilistic occupancy grid improvement map corresponds to surfaces that the IR projector of the Kinect cannot capture well with its IR camera, such as the car windows, rear light diffuser, and occlusion regions beneath and to the side of the car. Additionally, as seen in Scene 1, the maximum improvement also occurs on surfaces whose measurements gradually exceed the maximum supported depth of the probabilistic occupancy grid (5.0m). The negative improvement regions (Figure 5.37 d)) occur around the boundaries of regions, effectively outlining the regions of the car that could be acquired by the Kinect, as well as part of the pipe that is present in the top right hand third of the scene.



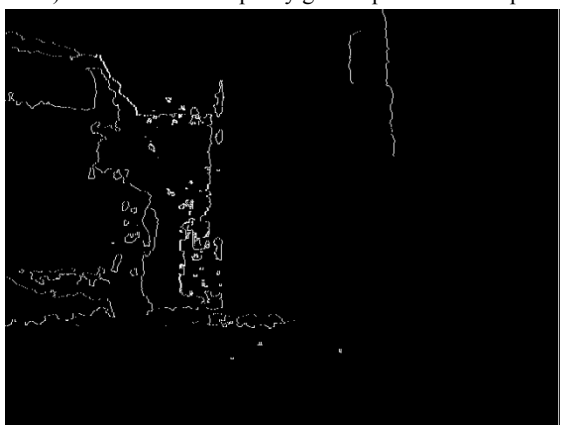
a) RGB image of scene



b) Probabilistic occupancy grid improvement map



c) High improvement regions (DP'_2)

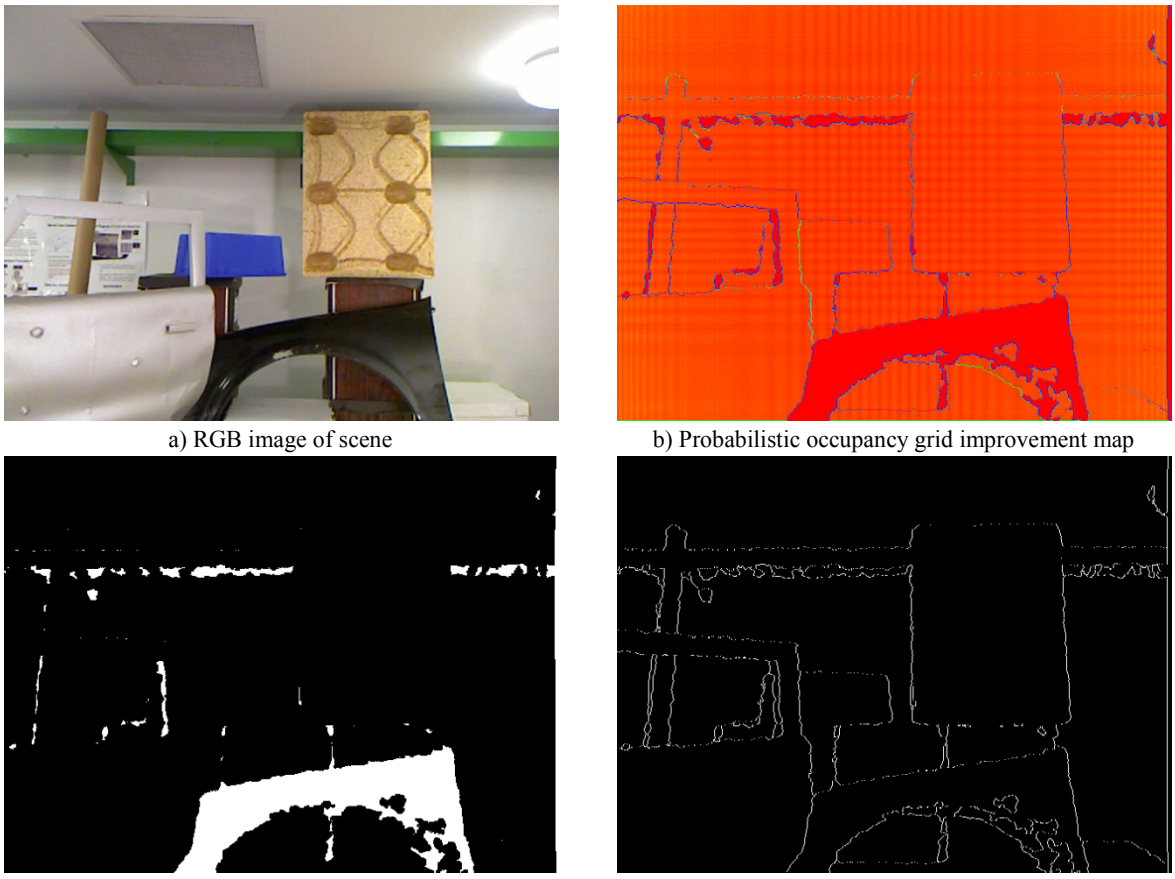


d) Negative improvement regions (DP''_2)

Figure 5.37 - Probabilistic occupancy grid improvement based selection processes on Scene 4

5.3.3.3 Scene 6 - Cluttered Scene in a Lab

Scene 6 of the cluttered lab, as presented in Figure 5.38 a), shows a front car wheel well panel in black that provides a surface that the Kinect cannot reliably capture values on, and hence this region is marked as high potential improvement (red), along with regions where the Kinect has occlusions introduced due to the separation between its IR camera and the IR projector, Figure 5.38 c). The negative improvement regions demonstrate a quite remarkable outlining of all the major objects in the scene (Figure 5.38 d)).



a) RGB image of scene

b) Probabilistic occupancy grid improvement map

c) High improvement regions (DP'_2)

d) Negative improvement regions(DP''_2)

Figure 5.38 - Probabilistic occupancy grid improvement based selection processes on Scene 6

5.3.3.4 Summary of Analysis

As illustrated in the results above, the high improvement regions obtained with the probabilistic occupancy grid improvement map correspond to locations that need to be acquired to get a better representation of the scene, although this may take several attempted acquisitions as many of these regions correspond to locations that the sensor has difficulty perceiving, or lie outside of the absolute range of the sensor. The negative improvement

regions provide a good measure of where transition regions lie. As a result, these locations should also be re-acquired by the sensor to ensure that they are reliably represented, provided that they are within the maximum depth of field of the sensor. Furthermore, these negative improvement region map can be used as guidance in a 3D segmentation algorithm, as they are effective at outlining edges of objects, provided that the transition is deep enough.

5.3.4 Discussion on the Probabilistic Occupancy Grid Improvement Based Selection Processes

Section 5.3.1 provides a quantitative analysis of the performance of the proposed probabilistic occupancy grid improvement based selection process, as compared against the random point selection process. This analysis led to the conclusion that both of these selection processes perform poorly at accurately capturing the scene, through the use of the surface mesh improvement metrics, but they both provide good coverage of the scene, with the probabilistic occupancy grid improvement based selection process providing better coverage, as exhibited by the probabilistic occupancy grid improvement metrics. This conclusion was verified in Section 5.3.2, which presented the progression of locations selected for acquisition, by the observations that although both selection processes studied choose points in a uniform density across the whole scene, the random point selection process will choose locations to acquire that are directly adjacent to locations already chosen, while the proposed probabilistic occupancy grid improvement based selection process does not. Using the probabilistic occupancy grid model created by the integration of the progressively selected points, the further verification that the surfaces of the objects in the scene are acquired at similar density can be made by observing the distribution of the voxels throughout the scene. The presence of empty spaces between the higher probabilities of occupancy voxels provide evidence that the probabilistic occupancy grid does not have any inherent procedure for interpolating over large unacquired areas, in addition to the poor accuracy in determining the boundaries of features.

As discussed previously, the probabilistic occupancy grid improvement based selection process does not perform well at accurately determining the boundaries of features, that is until the scene is fully acquired. This is explored in Section 5.3.3, which indicated that the probabilistic occupancy grid improvement based selection process is able to identify depth

transitions between different objects by selecting for negative improvement, as well as the edges between perceptible (acquirable) and imperceptible (unacquirable) regions of the scene. Using this property, it will be possible to use the probabilistic occupancy grid improvement measure to guide a 3D segmentation. Regions that need further acquisitions, due to occlusions or being imperceptible to the sensor can also be identified through the selection of high improvement regions.

The probabilistic occupancy grid improvement measure chooses points that are uniformly distributed with no immediate neighbours, but does not perform well at determining the accurate boundary of objects and other features within the scene. Due to these reasons, using a complementary technique, such as the surface mesh improvement measure presented in Section 3.1, in combination with the probabilistic occupancy grid improvement measure is desired to reduce the number of points acquired in a scene.

5.4 Combined Improvement Measure Selection Process Evaluation

The combined improvement measure selection process, which leverages the strengths of both the proposed surface mesh improvement measure that tends to drive acquisition of data to improve the accuracy of the scene (as shown in Section 5.2), and the proposed probabilistic occupancy grid improvement measure that tends to drive the acquisition to improve coverage of the scene (as shown in Section 5.3), is evaluated in this section. The scenes and sensors involved for the evaluation of the combined improvement measure selection process are listed in Table 4.2, with Table 4.1 detailing the sensor parameters. The procedure of evaluation is as follows:

i) An initial 10 x 10 uniform subsampling is performed on the data. Locations that do not have a data point are considered as being unacquirable, that is for one reason or another the sensor is unable to perceive and record a depth value.

ii) The two proposed improvement measures are calculated and then the selection process choses the next location to acquire ranged data. The acquisition of the point for the purposes of this evaluation is considered to take zero time (as previously discussed in Section 5.1), and any locations that do not contain valid range measurements are considered as unacquirable. This process repeats until all the acquirable locations within the sensor's field of view have been selected.

iii) The combined improvement measure selection process is compared against the random point selection process, the surface mesh improvement based selection process, and the probabilistic occupancy grid improvement based selection in the various analyses performed in the individual subsections.

The first subsection performs a comparative quantitative analysis of the four selection processes mentions in iii) above. The second subsection contains a comparative analysis detailing the progression of each of the four selection processes on a subset of the chosen scenes as acquired by the sensors listed in Table 4.2. The third subsection provides a brief analysis of the possible data savings possible using the proposed combined improvement measure selection.

5.4.1 Quantitative Analysis

The quantitative analysis presented in this section compares the various selection processes by comparing plots of the mean and the 95% confidence interval of the various metrics recorded for each scene aggregated by the sensors used for acquisitions. This is accomplished by first determining the individual curves for each of the four selection processes being compared for each acquired scene and sensor. From the sample deviation, s, and the mean, μ, calculated over the n scenes acquired by a particular sensor as listed in Table 4.2, the 95% confidence interval, $CI(0.95)$, as presented in [103] can be calculated through the application of the Student-t distribution, $t_u(n)$, as shown in eq. (5.2). Some of the 95% confidence region for a particular graph may be outside of the displayed scale of the graph, and hence are not visible. Additionally since only one scene was acquired with the Neptec LMS, no confidence interval is plotted for this sensor.

$$CI(0.95) = \left[\mu - t_{\left(1-\frac{1-0.95}{2}\right)}(n) \cdot \frac{s}{\sqrt{n}}, \mu + t_{\left(1-\frac{1-0.95}{2}\right)}(n) \cdot \frac{s}{\sqrt{n}}\right] \tag{5.2}$$

The mean trend and its confidence interval over the scenes acquired by a particular sensor are considered for the analysis in this section, as this captures the overall performance of the selection processes over a wide variety of scenes in a compact and efficient representation. The metrics that are investigated include the surface mesh mean estimated improvement value, the probabilistic occupancy grid mean estimated improvement value, and the surface mesh estimated improvement based on the resulting application of each of the four selection

processes plotted against both sample number and execution time. Furthermore, the surface mesh derived depth map's peak-signal-to-noise-ratio (PSNR) is calculated for the surface mesh improvement based, the combined, and the random point selection processes. The performance of the probabilistic occupancy grid improvement based selection process is not considered in the PSNR analysis, as this metric is sensitive to accurate representation of the scene. As was mentioned previously in Section 5.1, each plot is smoothed using a median filter that is 1% of the total number of acquirable locations in the acquisition, such that large magnitude outliers are removed, and the underlying trend is put into evidence. Each subsection details the analysis of the results for each parameter.

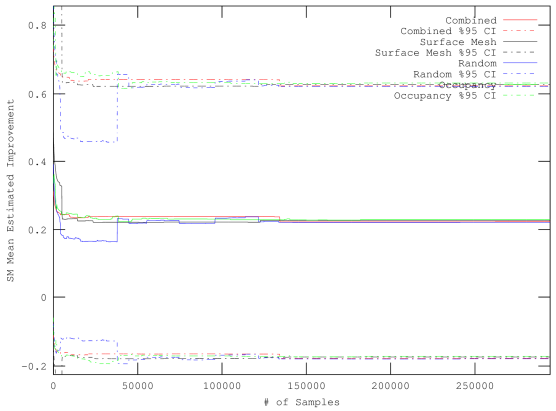
5.4.1.1 Surface Mesh Mean Estimated Improvement

The mean surface mesh improvement is calculated as the mean improvement value for the whole surface mesh improvement map at each step in the selection process. The aggregated results for each sensor over all applicable scenes are displayed in Figure 5.39 as plotted against the number of points chosen, and Figure 5.40 as plotted against the total execution time. Notice that in all the plots, the curves start out at high values, and eventually decay to their respective convergence values regardless of the sensor being employed (and as a result the sensor's native geometry). In situations where the sensor data produces significant outliers (e.g. the ASLS, subfigures b)), the techniques that provide more scene coverage, that is the random point selection process and the probabilistic occupancy grid improvement based selection process, provide faster convergence of this metric, likely due to the outliers relatively random placement in the scene. Additionally, the combined improvement measure and surface mesh improvement based selection processes have larger 95% confidence intervals in relation to random point selection and probabilistic occupancy grid improvement based selection when compared to the curves produced in other sensors. This implies that under the influence of outliers, the combined improvement and the surface mesh improvement bases selection processes exhibit more variation in this parameter. In the other cases where there are not as many outliers, both the surface mesh improvement based selection process and the combined improvement measure selection process provide more rapid convergence to the minimum values compared to random sampling, with the combined improvement measure selection process converging earlier than the surface mesh

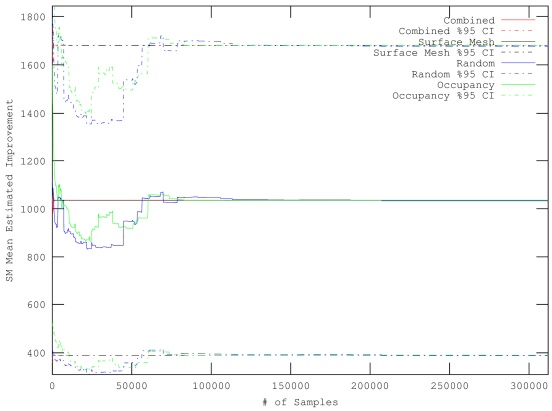
improvement based selection process, implying that the former selection process is better at driving acquisitions towards improvement scene accuracy.

In all cases, the combined improvement measure selection process provides slightly faster convergence in the time plots (Figure 5.40) compared to the surface mesh improvement based selection process. The random point selection process performs better than both of the surface mesh improvement dependant selection processes. On the other hand the probabilistic occupancy grid improvement based selection process provides earlier convergence for the higher resolution sensors (ASLS, Neptec) than the two surface mesh improvement dependant selection processes, but worse for the other lower resolution sensors. This is due to better coverage yielding faster to calculate smaller triangles earlier on in the acquisition process. Observations made on the time based graphs show an early plateau, more evident in the SICK/Track results (Figure 5.40 e)), in the curves representing the combined improvement measure and surface mesh improvement based selection processes. This occurs due to the fact that selecting a location which has no range data, yields no improvement, but it does have a real time cost. Random point selection does not have this penalty, as the choices made do not require any calculations, and the effects of this penalty are distributed more evenly through the acquisition time for the probabilistic occupancy grid improvement based selection, due to the focus on coverage.

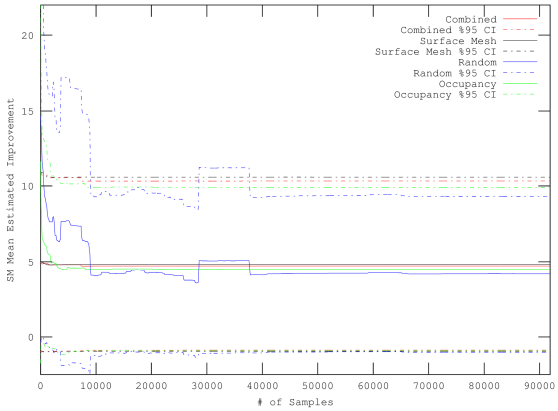
The same trend as observed in Section 5.2 holds true for the plots presented in Figure 5.39, namely that the mean surface mesh improvement value converges to its minimum after 5000-10000 points inserted, regardless of the sensor used and its corresponding native geometry, with the combined improvement measure and surface mesh improvement based selection processes providing a more reliable and stable convergence, when compared to the random selection process and the probabilistic occupancy grid improvement based selection process. The time to convergence varies according to sensor, due to the fact that the computational time is sensitive to the resolution used to generate the improvement maps, which is different for each sensor.



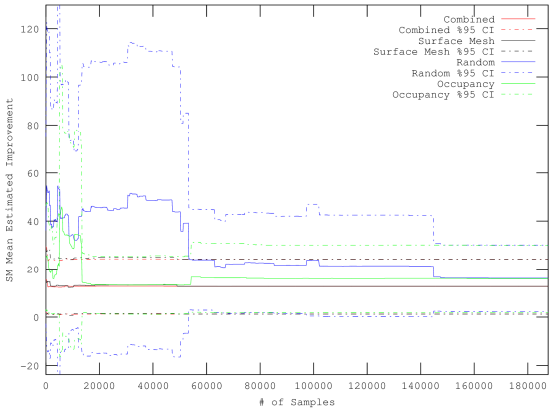
a) Kinect



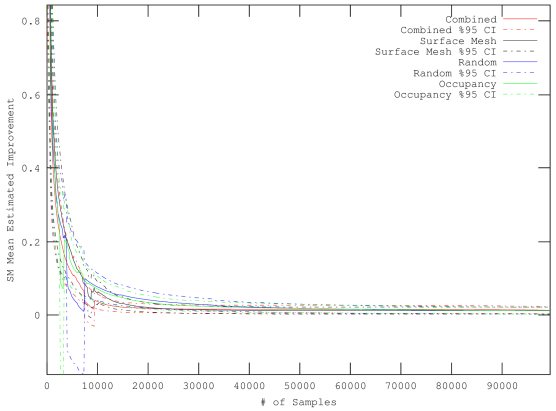
b) ASLS



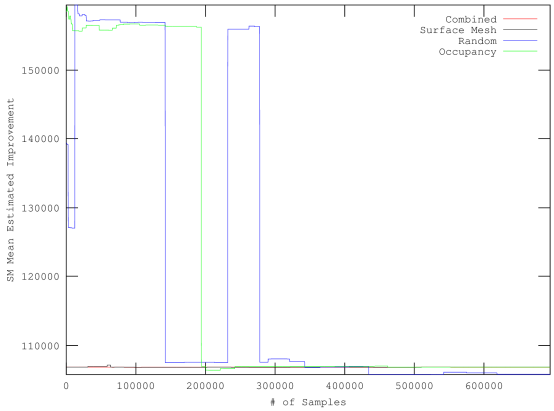
c) Jupiter mounted on F3T robotic arm



d) Jupiter mounted on linear track

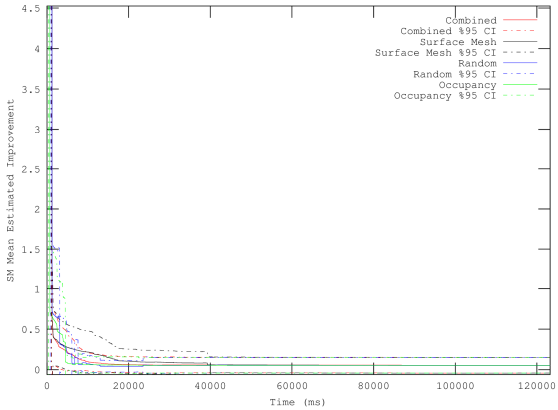


e) SICK LMS mounted on linear track

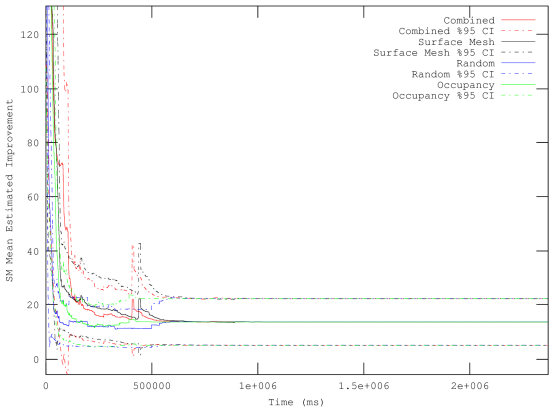


f) Neptec LMS

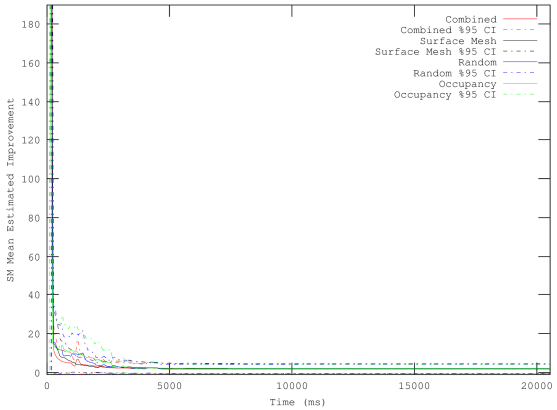
Figure 5.39 - Aggregate surface mesh mean estimated improvement vs. number of selected samples for various sensors



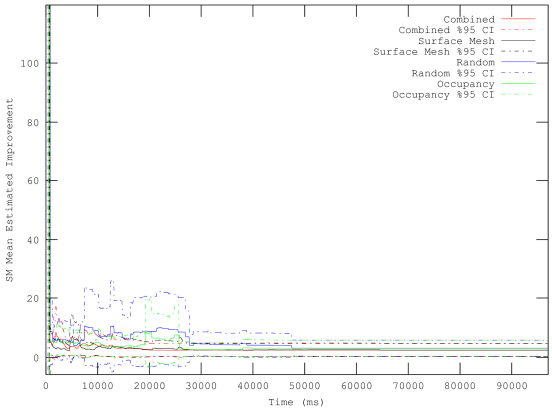
a) Kinect



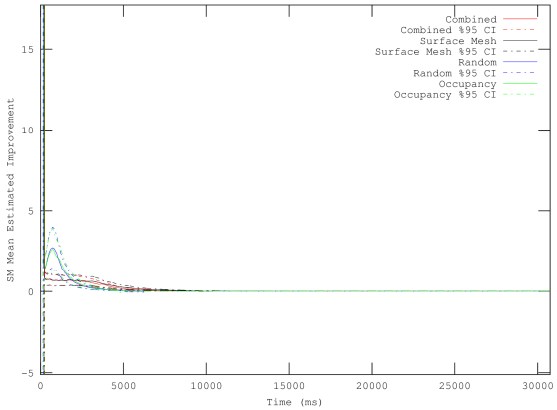
b) ASLS



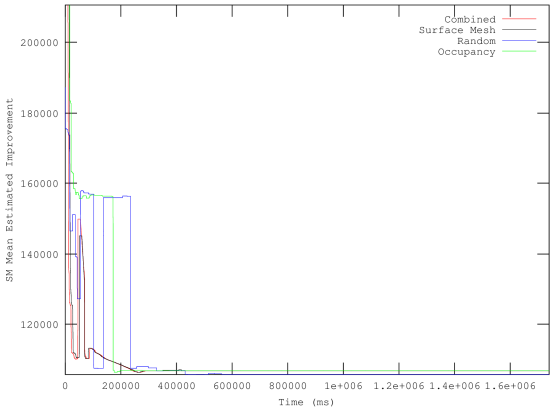
c) Jupiter mounted on F3T robotic arm



d) Jupiter mounted on linear track



e) SICK LMS mounted on linear track



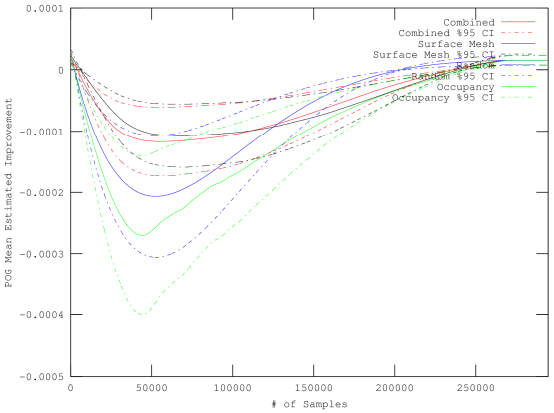
f) Neptec LMS

Figure 5.40 - Aggregate surface mesh mean estimated improvement vs. total execution time for various sensors

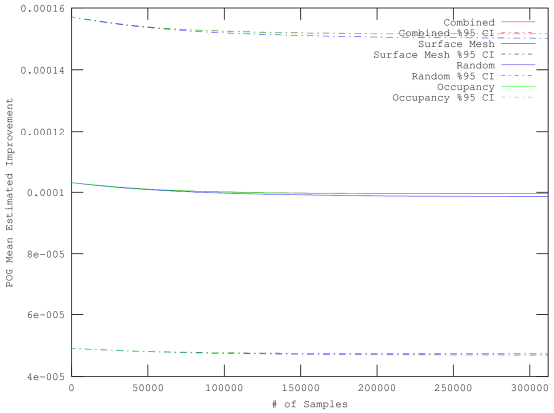
5.4.1.2 Probabilistic Occupancy Grid Mean Estimated Improvement

The mean probabilistic occupancy grid improvement metric provides the mean of the probabilistic occupancy grid improvement map for each sample selected by the respective selection processes. As discussed in Section 5.1, better performance in term of scene coverage, is represented by this metric by a larger initial decrease. Figure 5.41, which plots the probabilistic occupancy grid mean estimated improvement against the number of points acquired, indicates that the curve representing the combined improvement selection process lies in between the curve produced using surface mesh improvement based selection process and the curve produced using the probabilistic occupancy grid improvement based selection process. This indicates that, as expected by the design presented in Section 3.3.3, the combined improvement measure selection process provides better directed coverage of a scene than the surface mesh improvement based selection process, but not as much as the probabilistic occupancy grid improvement based selection process and the random point selection process. The degree of improved coverage introduced by the combination varies according the sensor employed, as this is due to the variation of the OPDF parameters between sensors, as well as their resolutions. In the plots of this metric for the ASLS (Figure 5.41 b)) and the Neptec LMS (Figure 5.41 f)), there is less evident difference between the selection processes, with the improved focus on coverage of the combined improvement selection process becoming better as the sensor resolution decreases towards the medium resolution sensors (Kinect shown in Figure 5.41 a) and the Jupiter mounted on the track in Figure 5.41 d)) and the lower resolution sensors (Jupiter mounted on the F3T robotic arm shown in Figure 5.41 c) and the SICK LMS mounted on the track shown in Figure 5.41 e)).

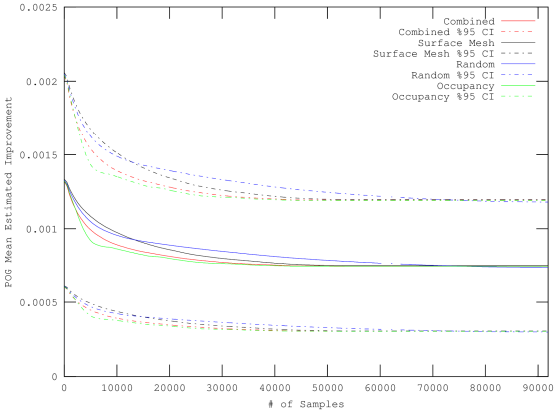
Observing the time based curves, Figure 5.42, random point selection reaches its minimum dip slightly earlier than probabilistic occupancy grid improvement based selection for the case of the Jupiter sensor acquisition, and slightly later for the other sensors. Each of these two selection processes reach their minimums much earlier than the combined improvement measure selection process, which in turn outperforms the surface mesh improvement based selection process, except for the case of the Neptec LMS sensor acquisition (Figure 5.42 f)). The narrower confidence interval for the random point and probabilistic occupancy grid improvement best selection processes indicates that they provide more consistent results across the variety of scenes with respect to time than the other two point selection methods.



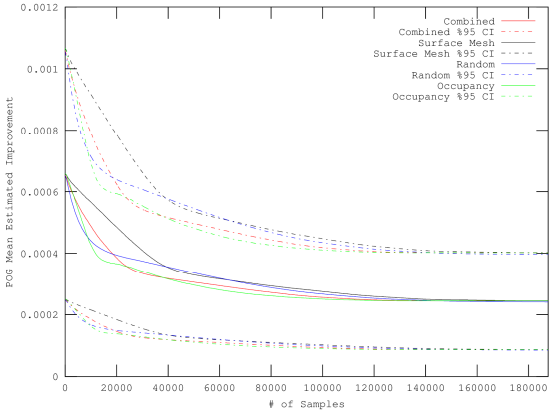
a) Kinect



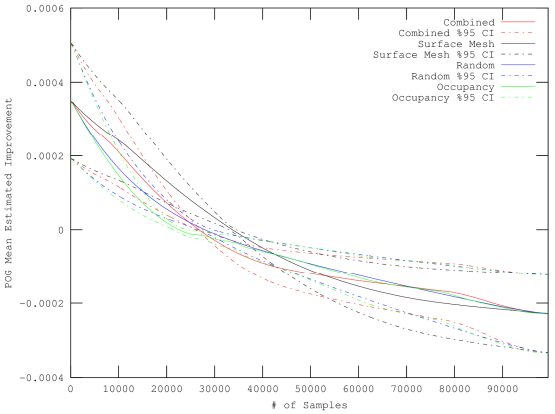
b) ASLS



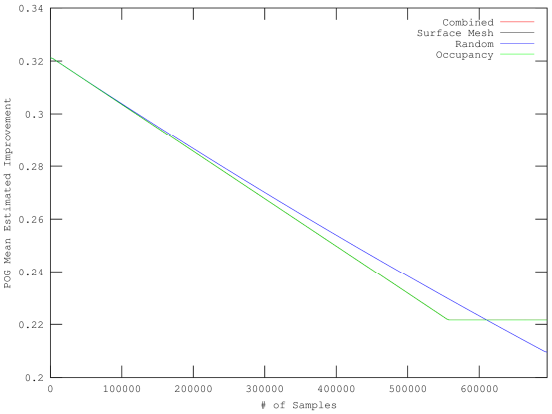
c) Jupiter mounted on F3T robotic arm



d) Jupiter mounted on linear track

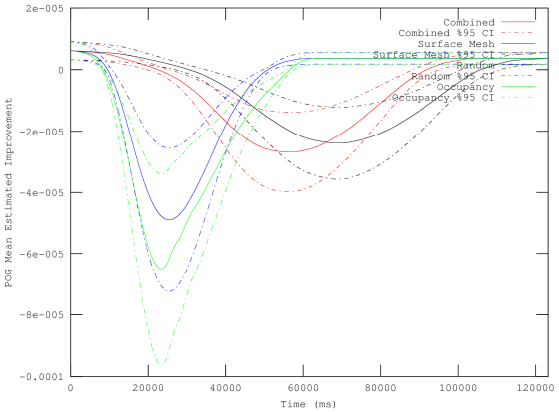


e) SICK LMS mounted on linear track

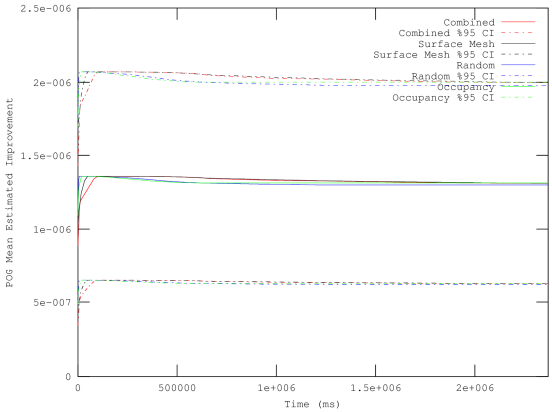


f) Neptec LMS

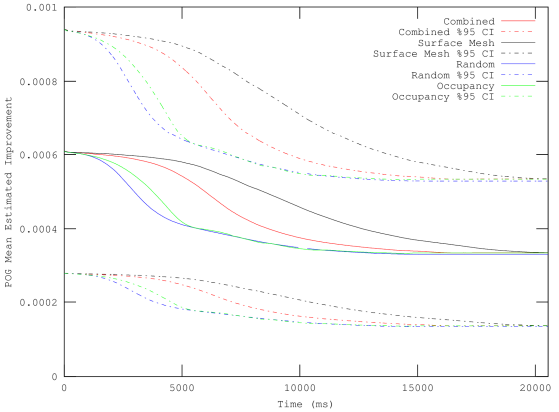
Figure 5.41 - Aggregate probabilistic occupancy grid mean estimated improvement vs. number of selected samples for various sensors



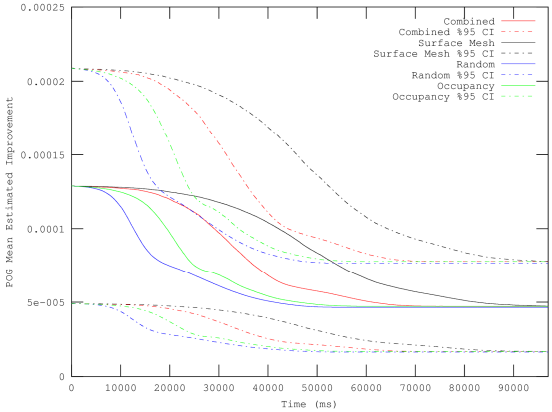
a) Kinect



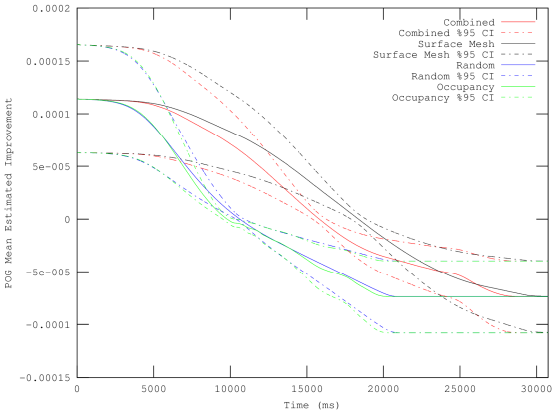
b) ASLS



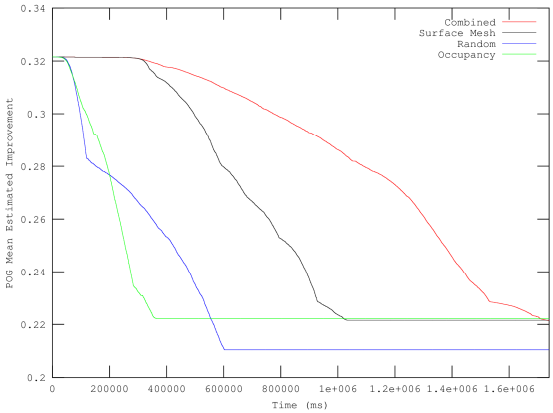
c) Jupiter mounted on F3T robotic arm



d) Jupiter mounted on linear track



e) SICK LMS mounted on linear track



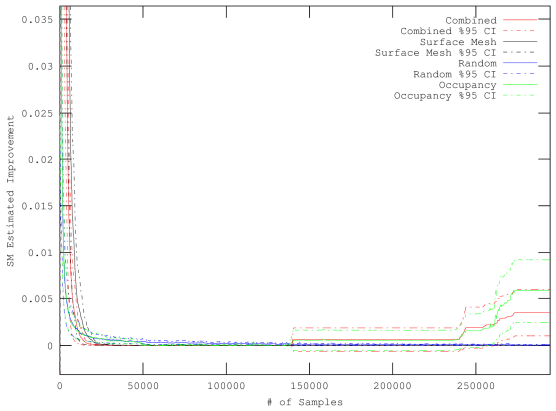
f) Neptec LMS

Figure 5.42 - Aggregate probabilistic occupancy grid mean estimated improvement vs. total execution time for various sensors

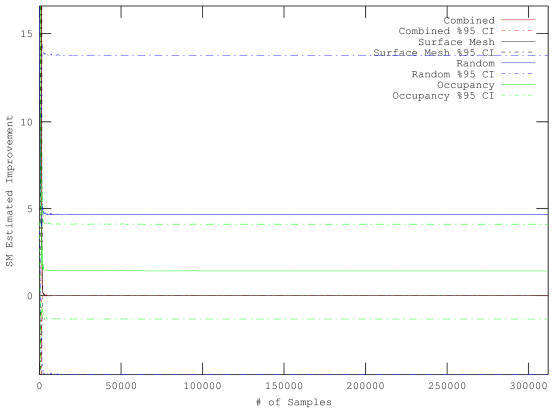
5.4.1.3 Surface Mesh Estimated Improvement

Surface mesh estimated improvement is the estimated improvement value of the next location to be added to the scene, and is a possible candidate to be used as a stopping criteria for the combined improvement selection process. As stated in Section 5.1, this metric is calculated inherently when determining the surface mesh improvement map, while the other metrics, such as the surface mesh and probabilistic occupancy grid mean estimated improvements, and the surface mesh and probabilistic occupancy grid estimated improvement errors, require much more additional time and computations to determine, and as a result, they are not optimal choices to use as a possible stopping criteria. As is visible from Figure 5.43, this value starts off high, and decays rapidly to its minimum value, after between 5000-10000 points captured, although the combined improvement selection process exhibits an explicit divergence when the scene is close to being fully captured. This holds true regardless of sensor and sensor geometry. This metric provides similar results to Section 5.4.1.1, but it is an inherently noisier metric as it is based on the estimated improvement values at specific point locations, where the surface mesh mean estimated improvement is calculated as the mean improvement over the entire improvement map for each point inserted, and as such some higher level filtering, such as was done when applying the median filter to generated the plots, is needed to obtain the underlying trends.

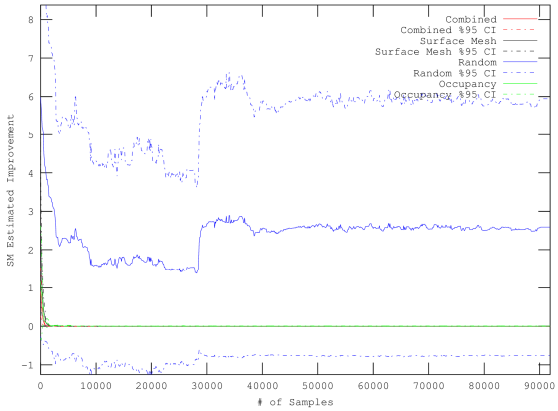
Figure 5.44, which illustrates the evolution of this metric with respect to execution time, exhibit similar trends as the surface mesh mean estimated improvement metric presented in Section 5.4.1.1, but with more chaotic and extreme variations as was the case with the point based results presented in Figure 5.43. The convergence of this metric, which represents the accurate modelling of objects within the scene, occurs early in the acquisition process. As a result many points that would otherwise be captured in a complete brute force acquisition can be left unacquired, as can be observed by the large stable plateaus in the plots of this metric for each sensor present in Figure 5.44.



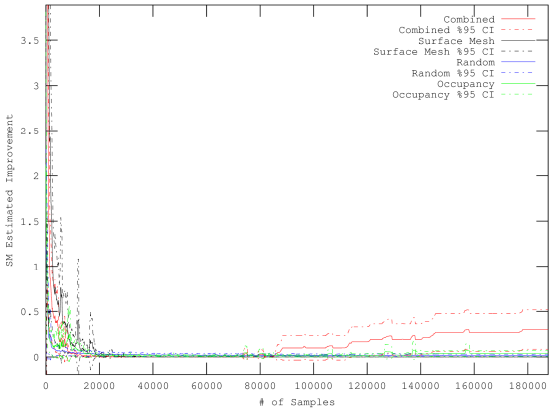
a) Kinect



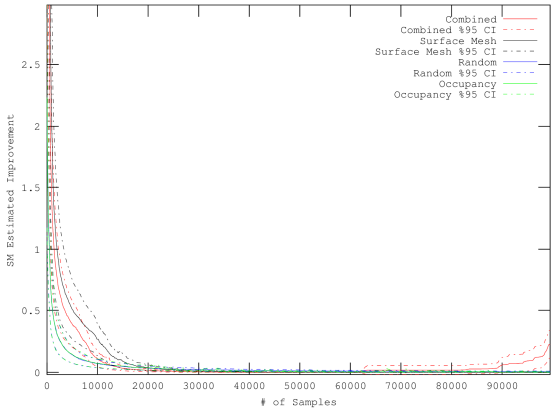
b) ASLS



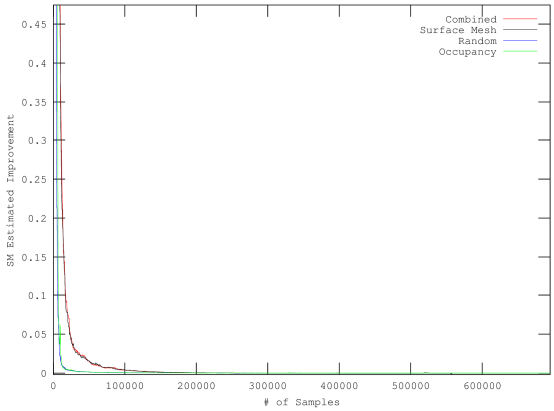
c) Jupiter mounted on F3T robotic arm



d) Jupiter mounted on linear track

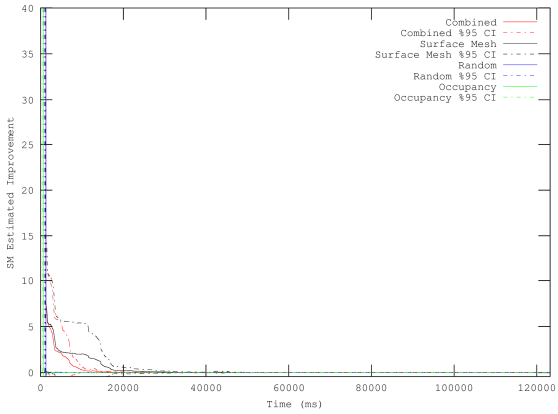


e) SICK LMS mounted on linear track

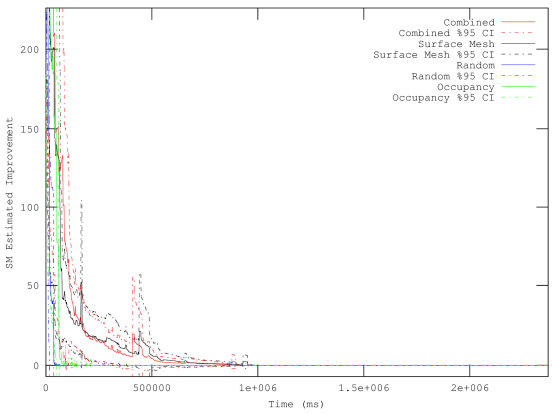


f) Neptec LMS

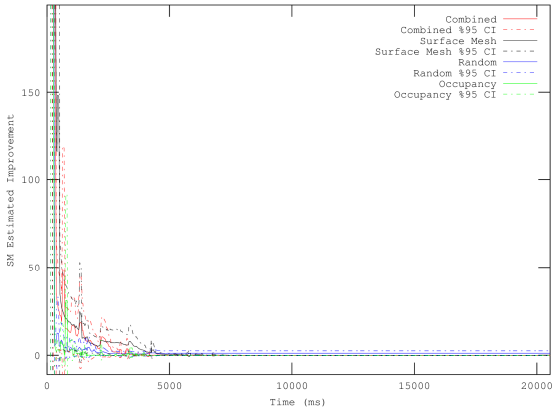
Figure 5.43 - Aggregate surface mesh estimated improvement vs. number of selected samples for various sensors



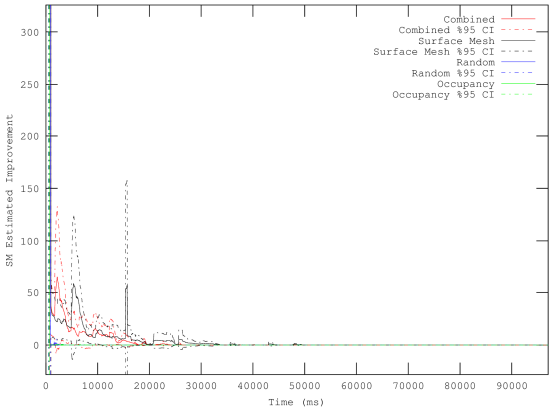
a) Kinect



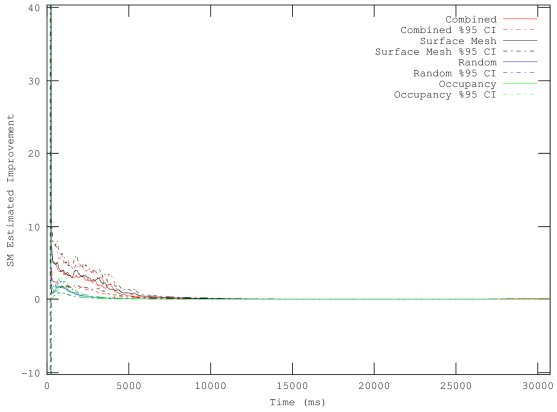
b) ASLS



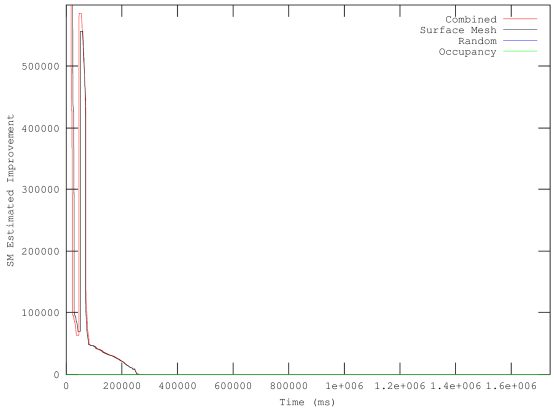
c) Jupiter mounted on F3T robotic arm



d) Jupiter mounted on linear track



e) SICK LMS mounted on linear track



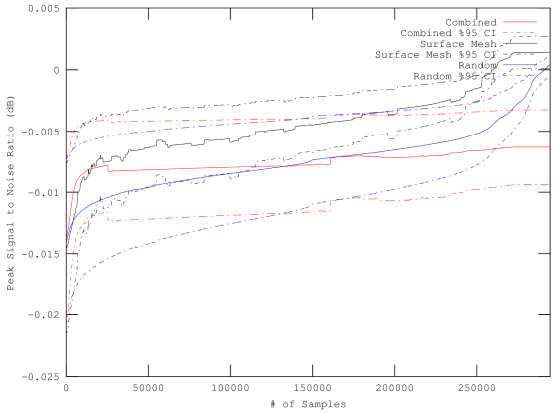
f) Neptec LMS

Figure 5.44 - Aggregate surface mesh estimated improvement vs. total execution time for various sensors

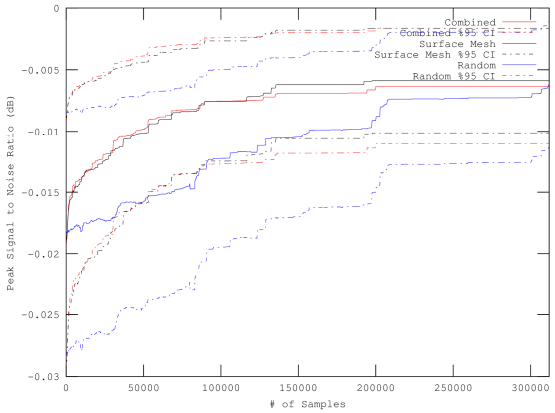
5.4.1.4 Estimated Surface Mesh Depth Map Peak Signal to Noise Ratio

The PSNR of the estimated surface mesh depth map, as discussed in Section 5.1, provides a measure of the quality of the estimated model of the scene. The probabilistic occupancy grid improvement based selection process is omitted from this comparison, as stated at the start of Section 5.4.1, as the PSNR depends highly on accuracy, and hence does not add significantly to the analysis presented in this section. A technique that provides a more accurate model of the scene for a given number of points, will exhibit a larger PSNR with fewer points inserted into the model, and hence, is a better point selection algorithm. Figure 5.45, which illustrates the aggregate PSNR against the number of points selectively inserted into the model, demonstrates that the surface mesh improvement based and the combined improvement selections processes achieve a superior PSNR with fewer points chosen than with random point selection. This verifies that they focus more on the accurate representation of the scene when compared to random point selection. Although in some of the subfigures (a) and c)), the surface mesh improvement based selection process may yield a better PSNR after 10000-50000 points have been chosen than the combined improvement selection process, the combined improvement selection process yields equivalent or superior performance in the previously identified region when most of the objects in the scene become accurately represented (5000-10000 points), regardless for sensor and geometry used.

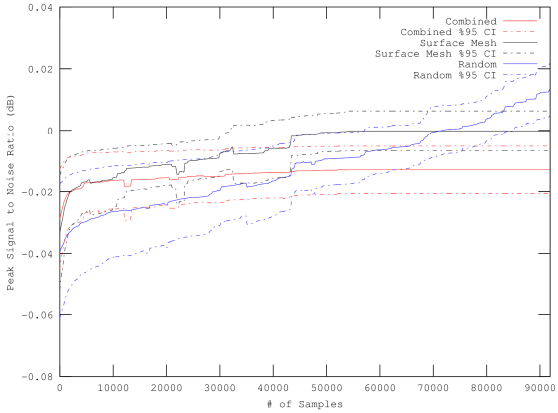
The analysis of the execution time against the PSNR, Figure 5.46, shows that the combined improvement measure selection process performs much better than the surface mesh improvement based selection process in the early stage of acquisition. The random point selection process exhibits similar PSNR levels to the combined improvement selection process until it nears full acquisition of the scene (observed as the plateauing on the maximum PSNR) in all the cases studied. The conclusion is that the combined improvement selection process does indeed outperform the random point selection process by maintaining a similar PSNR value early on in the acquisition process, but using significantly less points to achieve this. This further reinforces the assumption made in introduction of this thesis, that the number of points needed to accurately represent the scene is much less than a full acquisition of the scene.



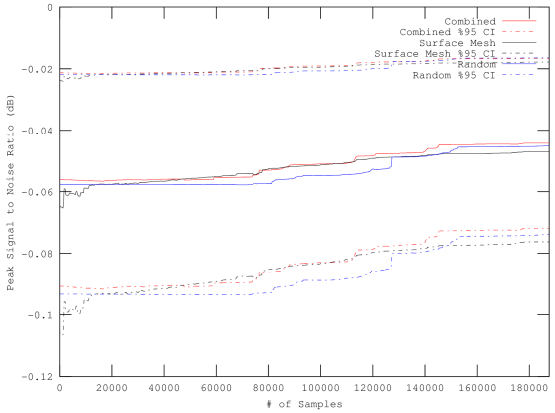
a) Kinect



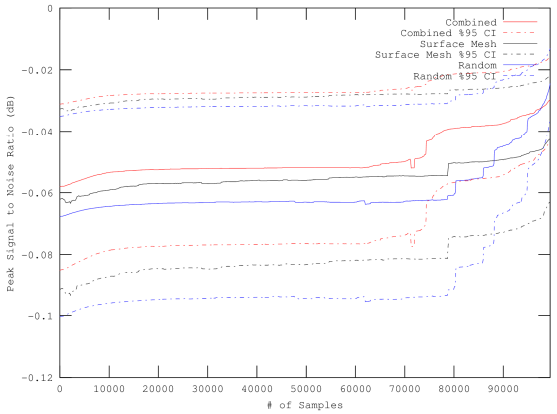
b) ASLS



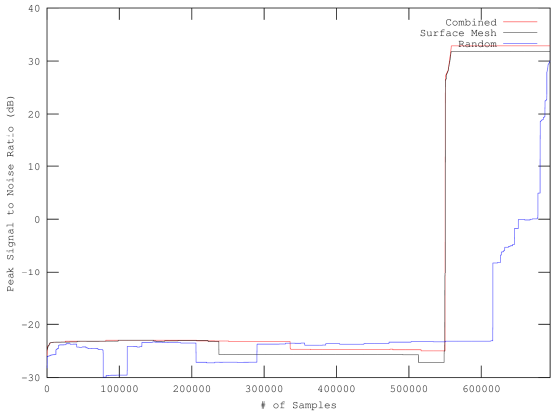
c) Jupiter mounted on F3T robotic arm



d) Jupiter mounted on linear track

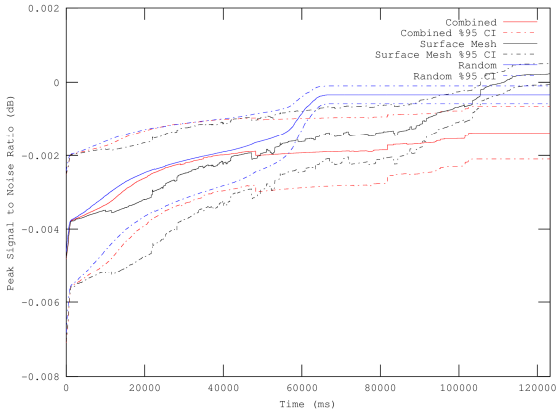


e) SICK LMS mounted on linear track

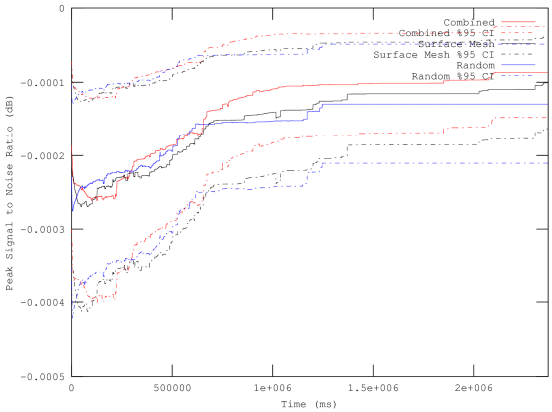


f) Neptec LMS

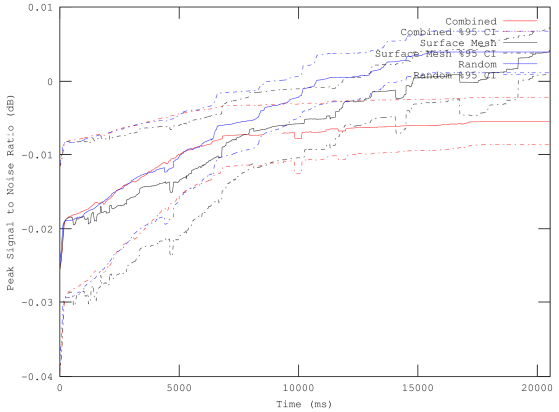
Figure 5.45 - Aggregate estimated surface mesh depth map PSNR vs. number of selected samples for various sensors



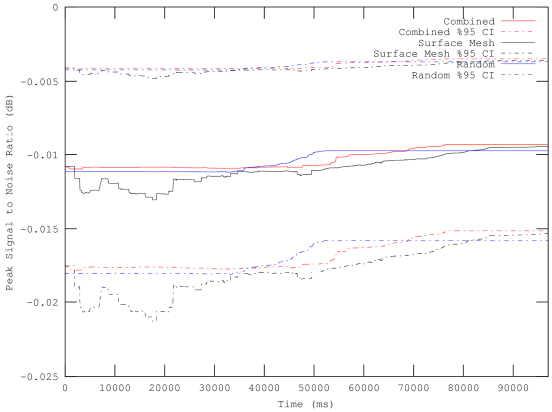
a) Kinect



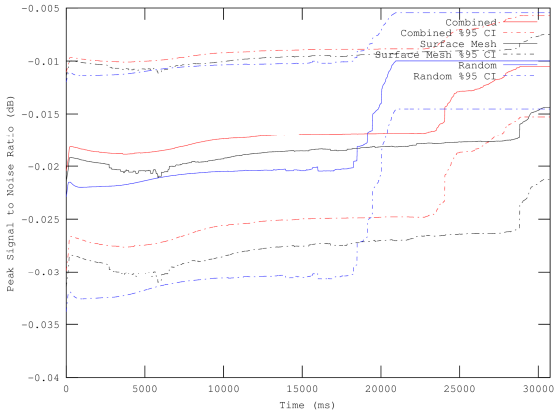
b) ASLS



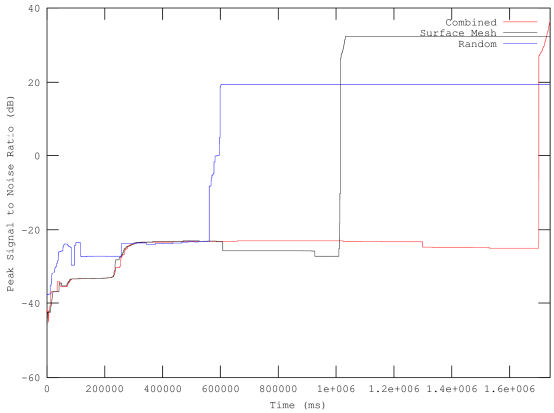
c) Jupiter mounted on F3T robotic arm



d) Jupiter mounted on linear track



e) SICK LMS mounted on linear track



f) Neptec LMS

Figure 5.46 - Aggregate estimated surface mesh depth map PSNR vs. total execution time for various sensors

5.4.1.5 Discussion of the Quantitative Performance

The experimental study conducted in this section puts in evidence that the combined improvement selection process provides faster convergence to the steady state parameter value compared to the surface mesh improvement based selection process for the mean surface mesh improvement measure (Section 5.4.1.1), and the surface mesh estimated improvement value (Section 5.4.1.3) across all sensors, encompassing various sensor geometries. This implies that it performs a better job of accurately capturing the scene geometry earlier in the acquisition process. Furthermore, the results shown in Section 5.4.1.2 indicate that the combined improvement measure selection process provides better coverage than the surface mesh improvement based selection process alone, thereby achieving the goal of amalgamating the benefits of both proposed improvement measures.

Additionally, as stated in Section 5.4.1.4, the combined improvement selection process provides better PSNR performance early in the acquisition process than random point selection and the surface mesh improvement based selection process for the number of samples added, and tracks the time based performance of random sampling. This implies that the combined improvement measure selection process provides a more accurate estimation of the depth map, than does random point selection, and furthermore, it has the capability, as designed, of reducing the number of points required to accurately represent the scene during acquisition time. If point acquisition time was not assumed to be zero during the application of the algorithms, then the combined improvement measure selection process has the capability to have far better performance than that of random sampling for slower sensors. This is due to the fewer points required to produce similar quality results yielding an overall savings in acquisition time.

5.4.2 Selection Process Progression Analysis

This subsection performs a comparative qualitative analysis of the selective progression of the acquisition of points, using the acquired point location maps previously used in Section 5.3.2, across a few selected scenes and sensors in order to demonstrate visually that the quantitative analysis matches the physical scene acquisition process. The selected scenes include Scene 8 of the exercise ball covered with a sheet, Scene 11 of the computer, Scene 13 of the fluidly deformed sheet, and Scene 18 of stacked boxes. The sensors that captured

these scenes, from which the progression results are being shown for are the Kinect, the ASLS, the Jupiter scanner mounted on the F3T robotic arm, the Jupiter scanner mounted on the linear motorized track, and the SICK LMS mounted on the linear motorized track. As the ASLS has much unacquirable space in its field of view, as discussed in Section 4.1.2, the resulting comparisons of acquired location maps have been cropped such that only the regions where objects are contained are visible.

The comparisons will be performed using a single map, with different coloured pixels representing locations chosen by the different selection processes. Red pixels correspond to locations chosen by only the combined improvement selection process, green pixels correspond to locations chosen by only the surface mesh improvement based selection process, blue pixels correspond to locations chosen by only the random point selection process, magenta pixel correspond to locations chosen by only the probabilistic occupancy grid improvement based selection process, and black pixels correspond to locations chosen by more than one selection process. The presentation of results obtained in Scene 8 of the exercise ball covered with a sheet, and Scene 18 of the stacked boxes is shown in this section. For further analysis on Scene 11 of the computer and Scene 13 of the fluidly deformed sheet see Section D.7.

5.4.2.1 Scene 8: Exercise Ball Covered with a Sheet

Scene 8, of an exercise ball covered with a white sheet (Section C.8) provides a large curved surface with a few large wrinkles. This yields a large depth transition between the covered ball, and the background wall for the larger field of view sensors, while still providing smaller scale features in the form of wrinkles. Figure 5.47 shows the comparison of points acquired of the scene, as captured by the Kinect for 2000 and 5000 selected points that have been inserted into the models of the scene. Figure 5.47 a) illustrates that the combined improvement selection process (red) has much better coverage than the surface mesh improvement based selection process (green) after 2000 points selected, as there are few places where locations have been chosen directly adjacent to each other and those are concentrated on the dominant contours of the ball. The combined improvement measure selection process has most of the ball outlined at this stage, as well as started investigating some of the surfaces of the objects. The randomly selected points (blue) and probabilistic

occupancy grid improvement based (magenta) selection process both have better coverage than the other two. By the time 5000 points have been selectively added (Figure 5.47 b)), the outer edge of the ball has been acquired with fairly high density by the surface mesh improvement based selection process (green) and the combined improvement selection process (red), but the latter process has acquired much more of the base that the ball is sitting on, much more of the wall, and much more of the surface of the ball, demonstrating a better balance between attraction to detailed features and coverage of the entire scene.

Figure 5.48 demonstrates the point selections made by the four selection processes based on the ASLS acquisition of the covered exercise ball. After the selective insertion of 2000 points (Figure 5.48 a)), the regions around the outside of the ball, along the wall, along the edges of the base, and on the surface of the ball have been sparsely selected by the combined improvement selection process (red). The surface mesh improvement based selection process (green), on the other hand, has acquired these regions using more points, but in less areas. The probabilistic occupancy grid based improvement selection process (magenta) has selected points relatively close together, while the random point selection process (blue) has chosen points that are uniformly distributed, which is further demonstrated after 5000 points are acquired (Figure 5.48 b)).

Figure 5.49 demonstrates the point selections made by the four processes when using the Jupiter scanner mounted on the F3T robot for acquiring the same scene. After 1000 points (Figure 5.49 a)) the combined improvement selection process (red) has more of the surface transitions acquired than does the surface mesh improvement based selection process (green). After 3000 points (Figure 5.49 b)), however, both methods have chosen points about the major features. The random point selection process has chosen more points in the center of the image, as seen by the higher density of blue pixels in this region, which becomes denser from 1000 points inserted (Figure 5.49 a)) until 5000 points are inserted (Figure 5.49 c)). The probabilistic occupancy grid improvement based selection process (magenta) has chosen points equally distributed through the sensor field of view, providing more coverage of the scene.

The progression comparison of the four selection processes on choosing locations to acquire from the scan of the covered ball made with the Jupiter mounted on the linear motorized

track, which only contains the ball and not the background surface (see Figure C.8 e)), is shown in Figure 5.50. After 2000 points (Figure 5.50 a), the combined improvement selection process (red) captured more coverage of locations along the edge of the ball, than the surface mesh improvement based selection process (green). Both of these have also selected points along vertical regions where particular scans where noisy. The probabilistic occupancy grid improvement based selection process (magenta) has chosen points that have horizontal line patterns, while being distributed relatively far apart vertically and covering most of the ball surface. The random selection process (blue) appears to be distributed uniformly. These trends continue when 5000 points have been selectively inserted (Figure 5.50 b)) with the combined improvement selection process choosing points which encompass all the major features as acquired by the sensor, including the lines of vertical noise, and the edge of the ball, but still providing decent coverage of the ball surface.

Figure 5.51 depicts the progression analysis of the points selectively chosen from the covered exercise ball scene acquired with the SICK LMS mounted on the linear motorized track for a) 1000 points, b) 3000 points, and c) 5000 points. The first observation that can be made over all three progression stages is that the surface mesh improvement based selection process (green) has chosen points that are adjacent to other points, while the combined improvement selection process (red) has chosen points that are more widely distributed. Furthermore points chosen by these two methods have formed along vertical lines, due to the differences in effective resolution between the two axes, namely the vertical axis covers a larger area with a lower sample rate than does the horizontal axis, thereby providing a greater uncertainty along this dimension, as well as the compressed vertical appearance of the scene. As the progression proceeds form 1000 points to 5000 points, the top of the ball (approximately along the line formed by top third of the field view) becomes heavily chosen by these selection processes, as well as the bottom of the ball is selected (the vertical chosen located along the bottom quarter of the field of view).

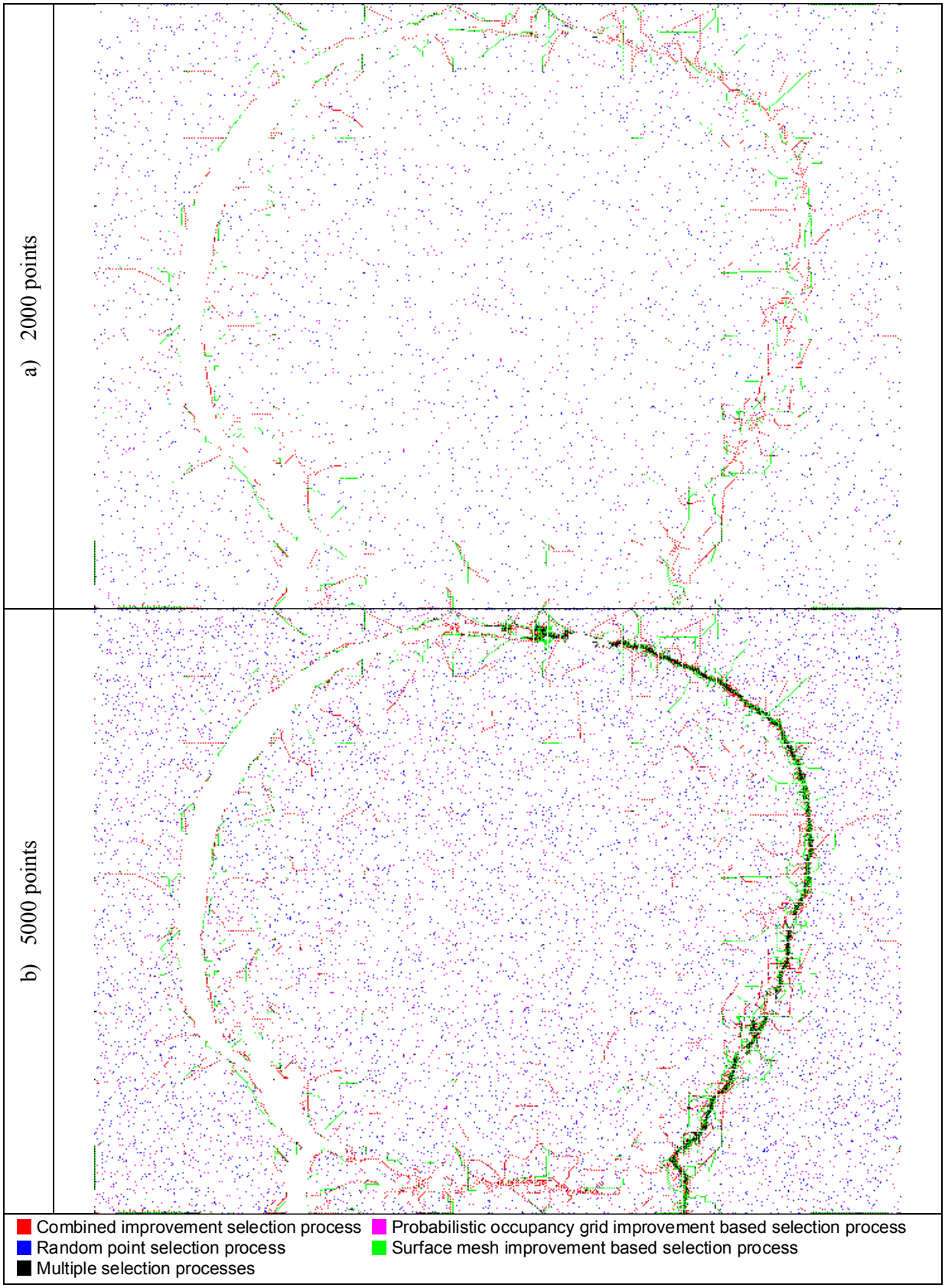


Figure 5.47 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the Kinect on the covered ball scene (#8)

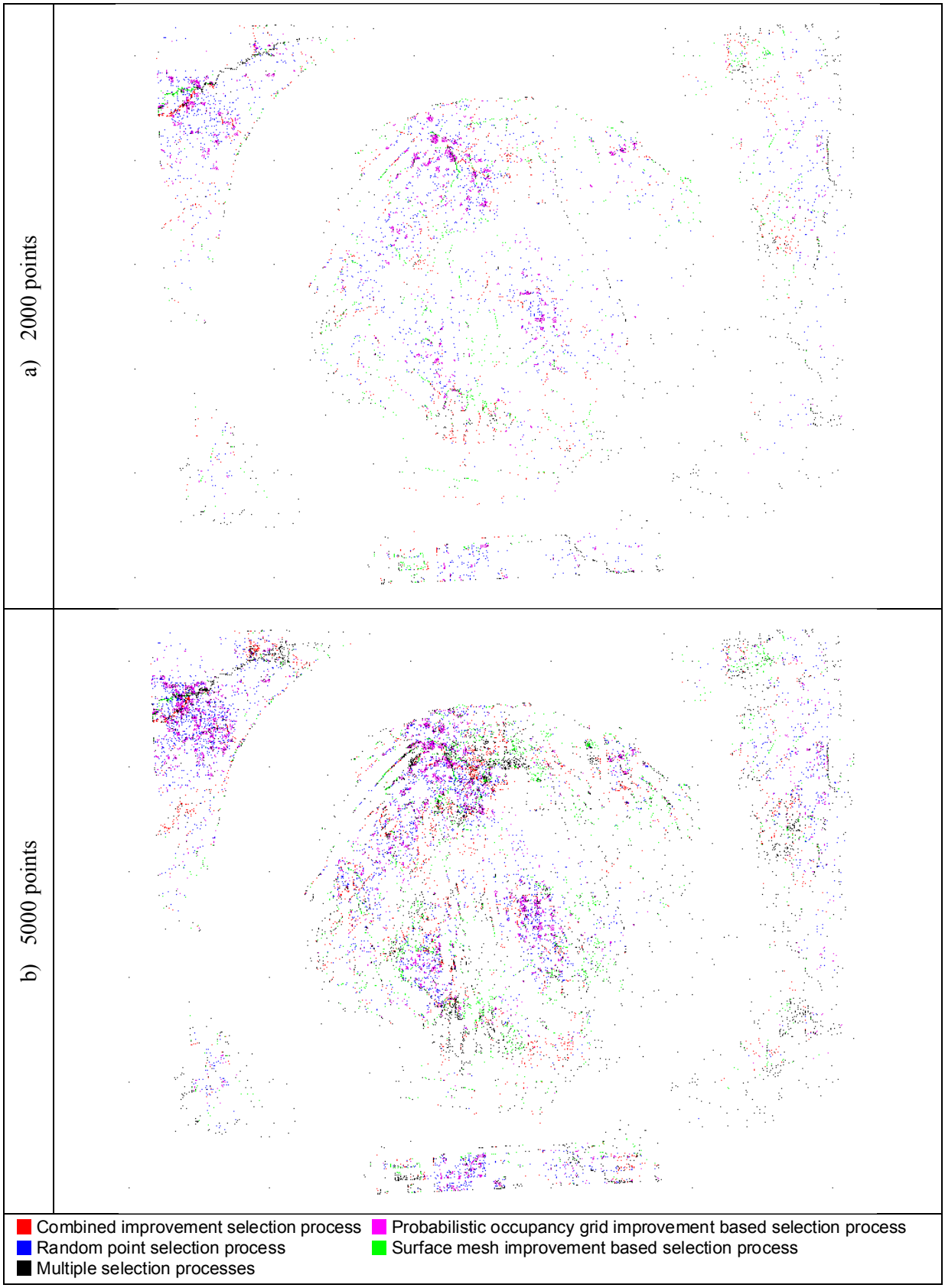


Figure 5.48 - Comparative progression for a) 2000 points and b) 5000 points selectively acquired with the ASLS on the covered ball scene (#8)

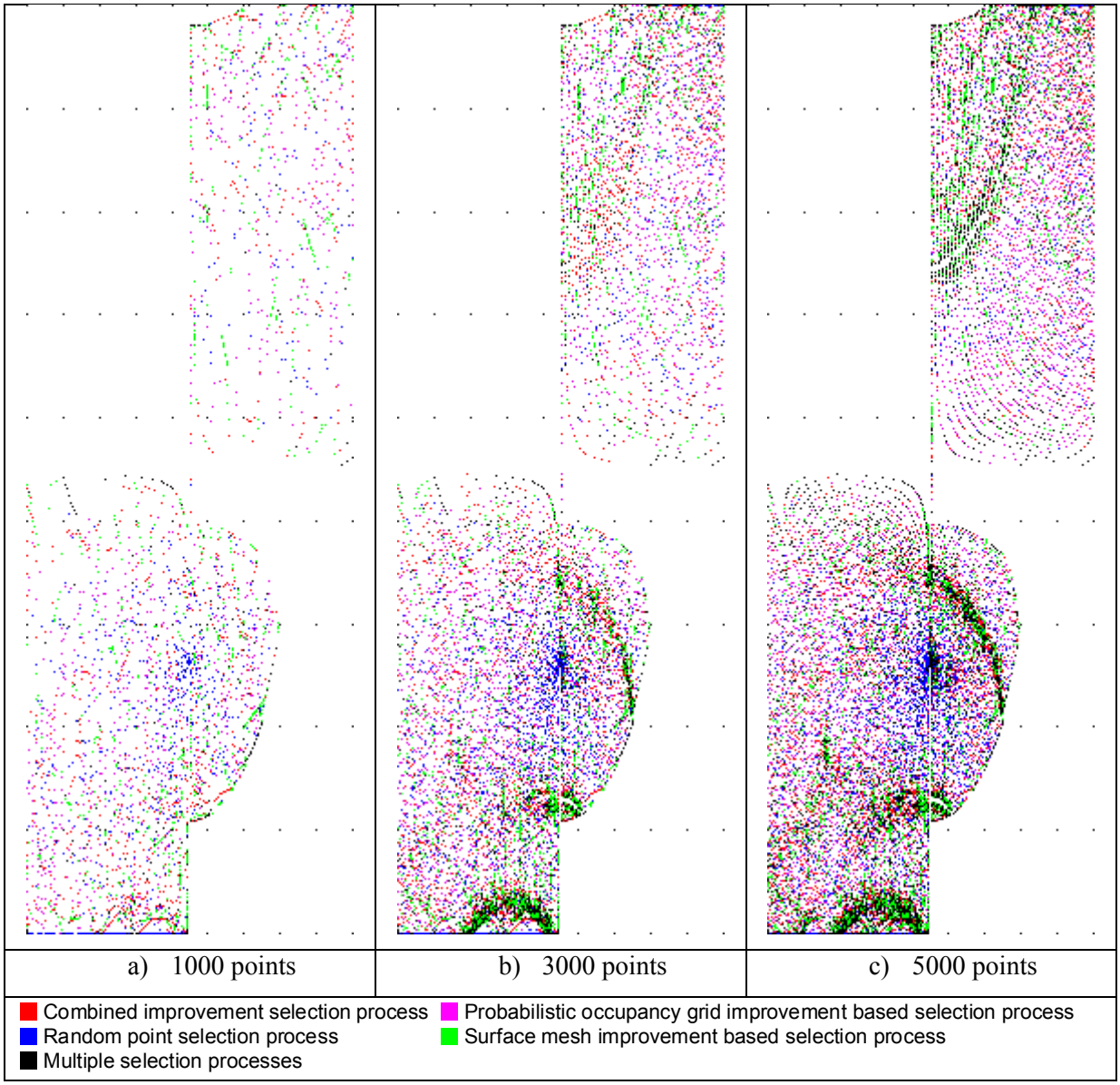


Figure 5.49 - Comparative progression for a) 1000 points, b) 3000 points, and c) 5000 points selectively acquired with the Jupiter mounted on the F3T robot on the covered ball scene (#8)

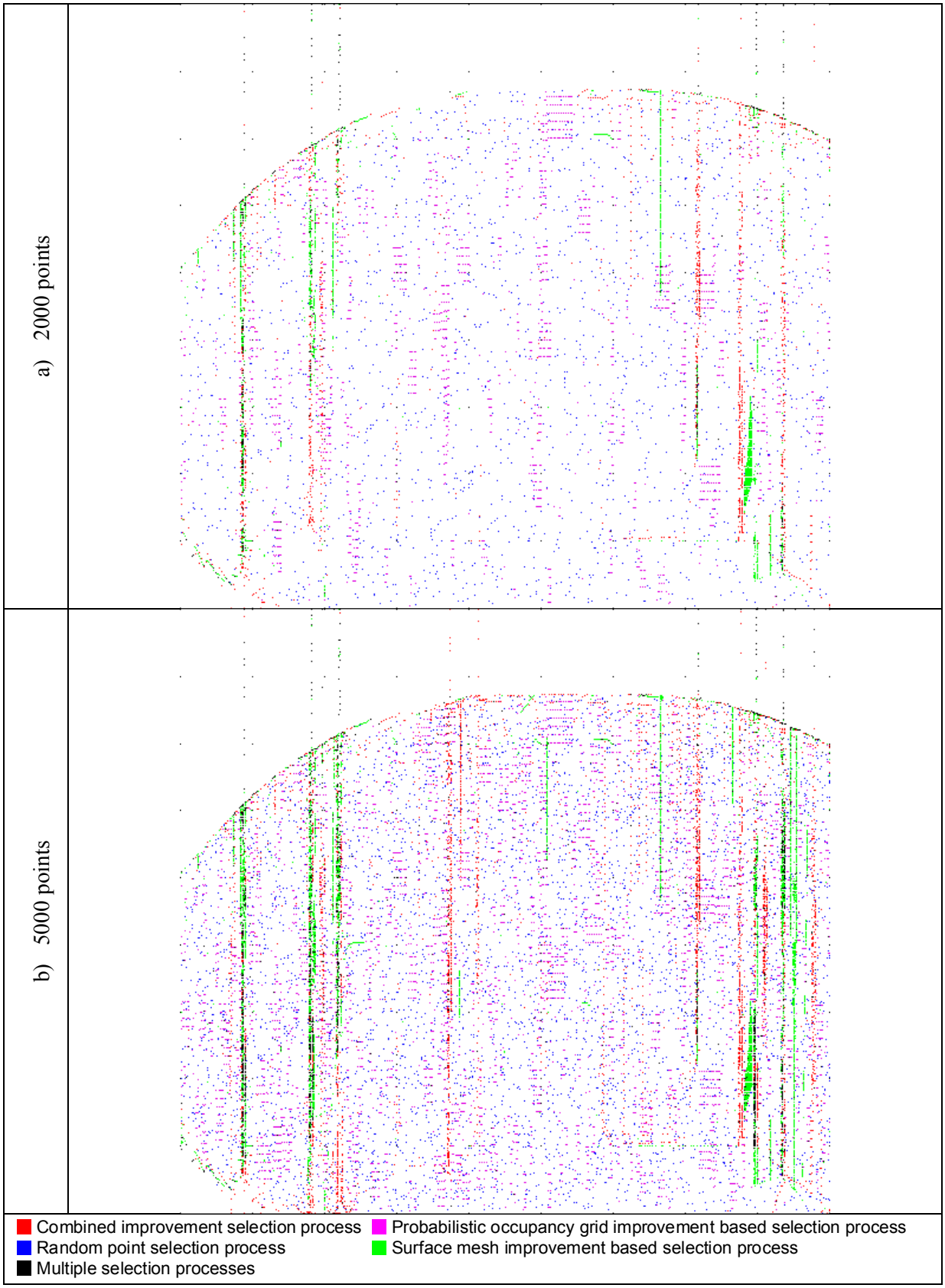


Figure 5.50 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the Jupiter mounted on the linear motorized track on the covered ball scene (#8)

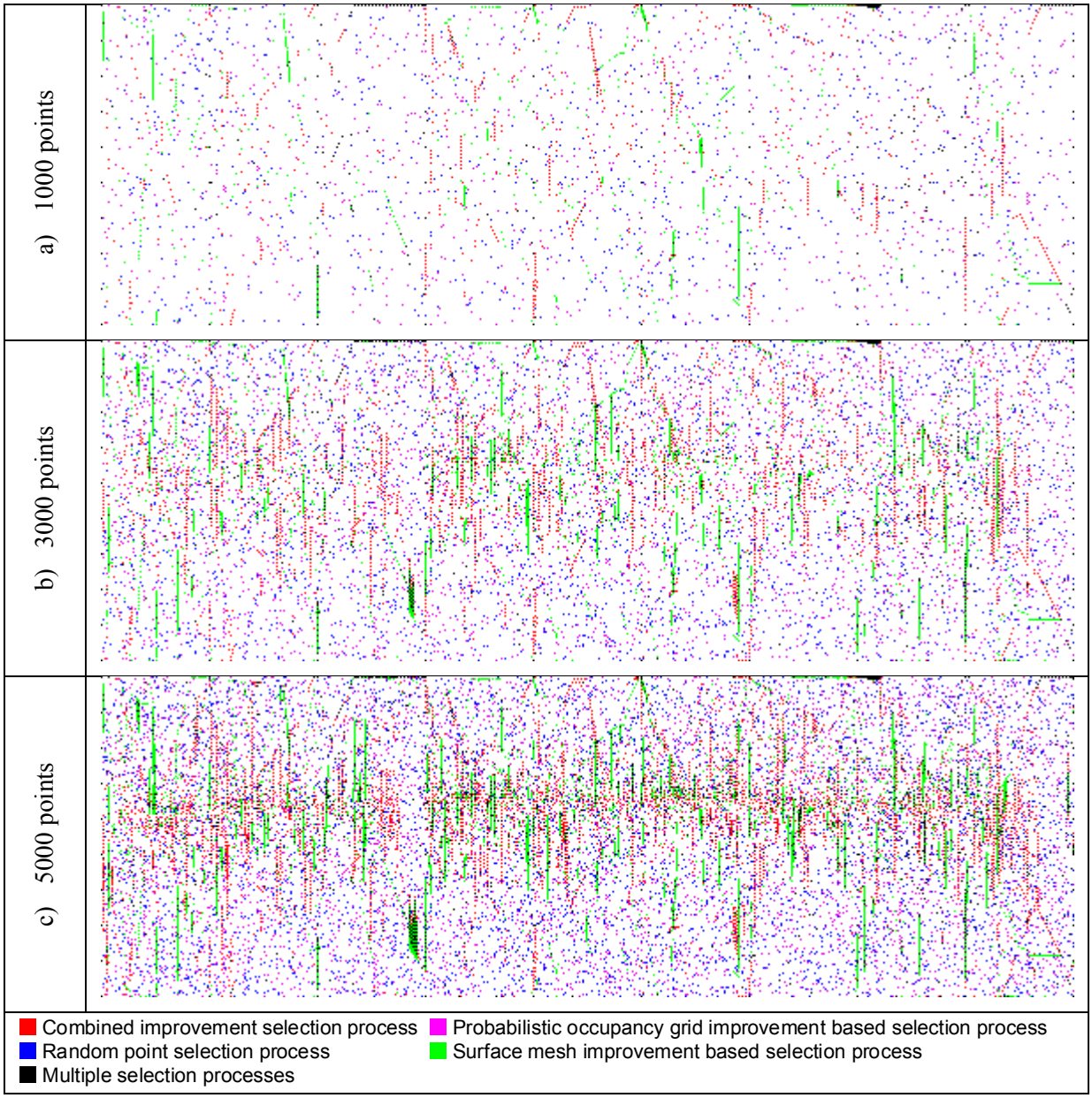


Figure 5.51 - Comparative progression for a) 1000 points, b) 3000 points, and c) 5000 points selectively acquired with the SICK LMS mounted on the linear motorized track on the covered ball scene (#8)

5.4.2.2 Scene 18: Stacked Boxes

Figure 5.52, which presents the comparison progression maps of the locations chosen by the selection processes for the stacked boxes scene (Scene 18) as acquired by the Kinect, puts into evidence that the combined selection process (red) chooses points that cover more edges and transitions in the scene than does the surface mesh improvement based selection process (green), regardless of the number of point chosen. By the time 5000 points, Figure 5.52 b), are acquired the deepest transition present in the scene (top and right of the stack boxes) has been densely selected by these two selection processes, with the combined improvement selection process having chosen more points along the other transitions. The random point selection process (blue) and the probabilistic occupancy grid improvement based selection process (magenta) have for each scenario (2000 points shown in Figure 5.52 a) and 5000 points shown in Figure 5.52 b)) chosen locations which uniformly cover the scene.

The comparison of the progression of point locations chosen by the four selection processes on the stacked boxes scene as acquired by the ASLS is shown in Figure 5.53. When only 2000 points have been chosen (Figure 5.53 a)), the combined improvement selection process (red) has covered more transitions present in the scene than the surface mesh improvement based selection process (green), especially in the center. After 5000 points have been added (Figure 5.53 b)), the trend is even more apparent, with the combined improvement selection process choosing locations on more edges and transitions in the scene than the surface mesh improvement based selection process. The probabilistic occupancy grid improvement based selection process (magenta), as evidenced in each case, chose points that are clustered, but with the clusters distributed throughout the scene. The random selection process (blue), on the other hand, has chosen locations to acquire that cover the scene uniformly.

The maps illustrating the comparative progression of the location choices made by the four selection processes on the stacked boxes scene as acquired by the Jupiter sensor mounted on the F3T robotic arm are shown in Figure 5.54. As the points chosen increase from 1000 points in Figure 5.54 a), to 5000 points in Figure 5.54 c), the combined improvement selection process (red) covers more transitions and edges with fewer points inserted, than does the surface mesh improvement based selection process (green), while also better preserving the distribution of points throughout the scene. The random point selection

process (blue) and the probabilistic occupancy grid improvement selection process (magenta), have chosen location which uniformly cover the scene and did not concentrate much on features.

Figure 5.55, which illustrates the comparative progression of the locations chosen by the four selection processes of the scene consisting of stacked boxes as acquired by the Jupiter mounted on the linear motorized track, shows that regardless of the number of points chosen, the random point selection process (blue) acquired points that cover the scene uniformly, while the probabilistic occupancy grid improvement based selection process (magenta) choose locations that form dotted line patterns that uniformly cover the scene. The combined improvement selection process (red), and the surface mesh improvement based selection process (green) focused heavily on identifying vertical noise present in the scene. This is to be expected because during acquisition, whether a measurement is erroneous or not cannot be known during acquisition.

Finally, the maps illustrating the comparative progress of the selection of point locations made by the four selection processes on the stacked boxes scene as acquired by the SICK LMS mounted on the linear motorized track are shown in Figure 5.56. Remember that the vertical axis appears compressed, due to the much lower sampling rate and larger field of view along this axis than the horizontal axis, hence the stacked boxes appear only along the middle third of the vertical axis. The progression of locations chosen by the combined improvement selection process (red) focuses on the deepest transition in the scene produced by the top of the stacked boxes and the wall, while also capturing some of the vertical transitions between boxes. The surface mesh improvement based selection process (green) focuses on the same features, but does so with more connected points per region. The other two selection process choose point locations that uniformly cover the scene regardless of how many points have been acquired.

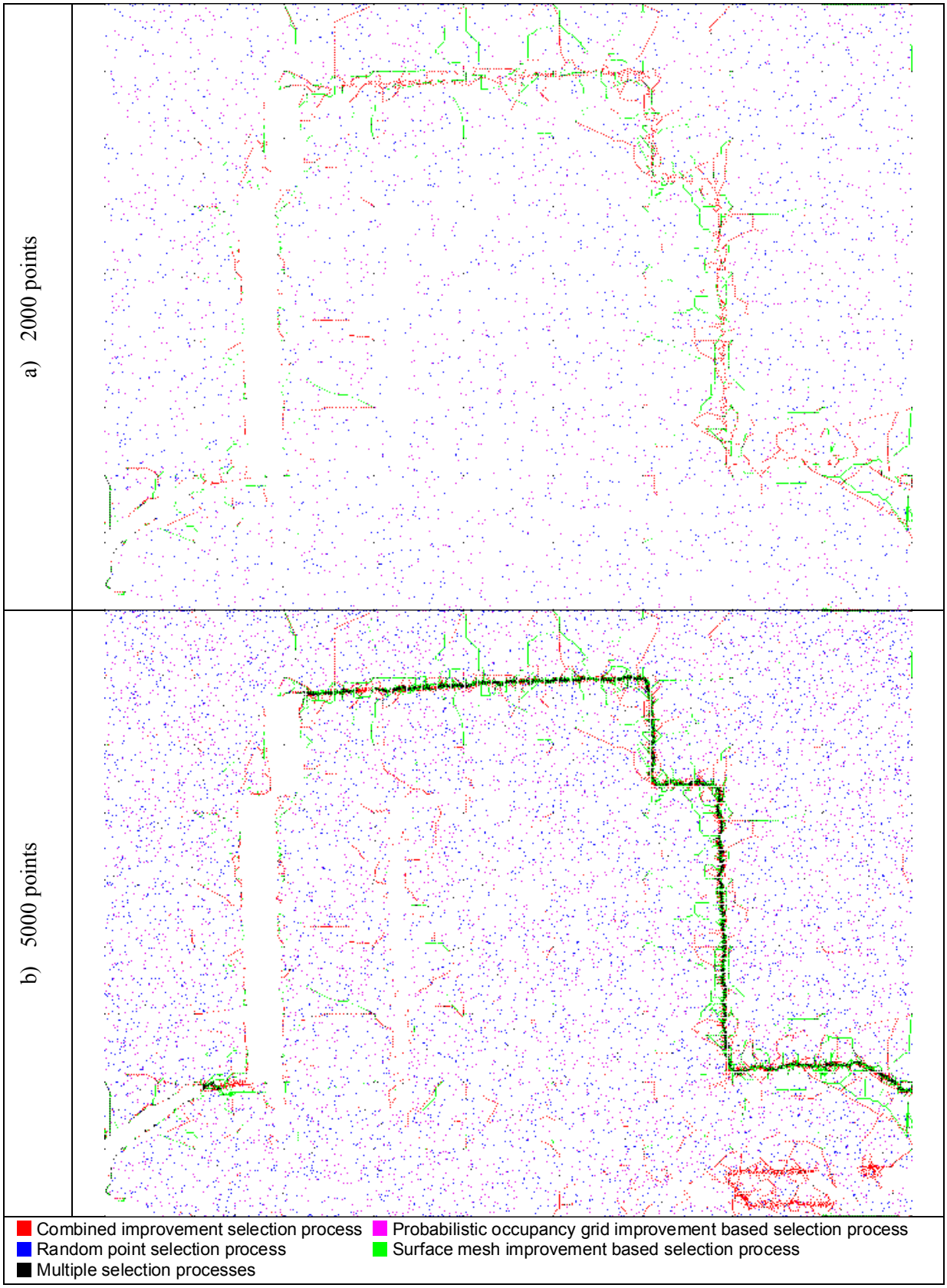


Figure 5.52 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the Kinect on the stacked boxed scene (#18)

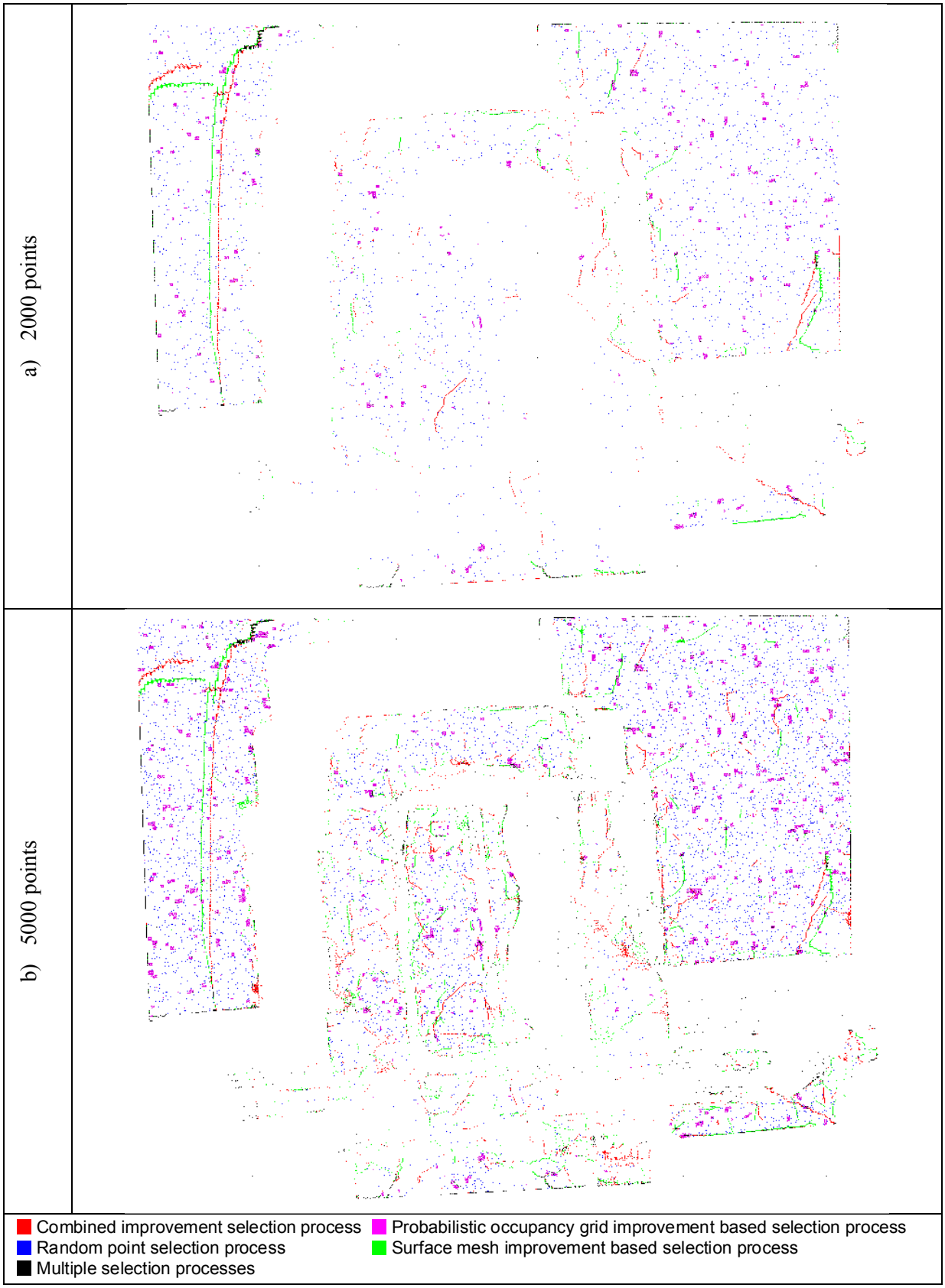


Figure 5.53 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the ASLS on the stacked boxed scene (#18)

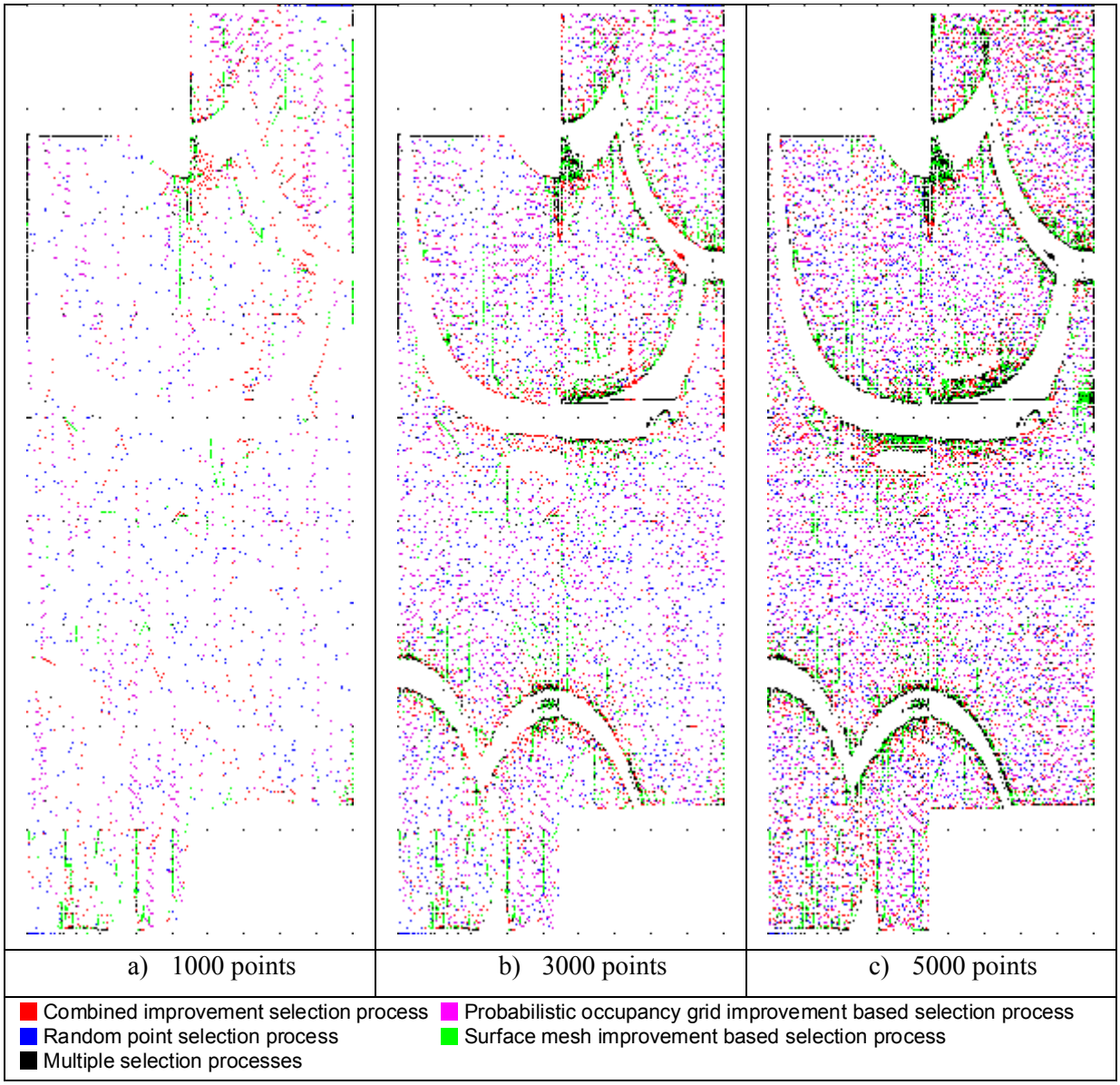


Figure 5.54 - Comparative progression for a) 1000 points, b) 3000 points, and c) 5000 points selectively acquired with the Jupiter mounted on the F3T robot on the stacked boxed scene (#18)

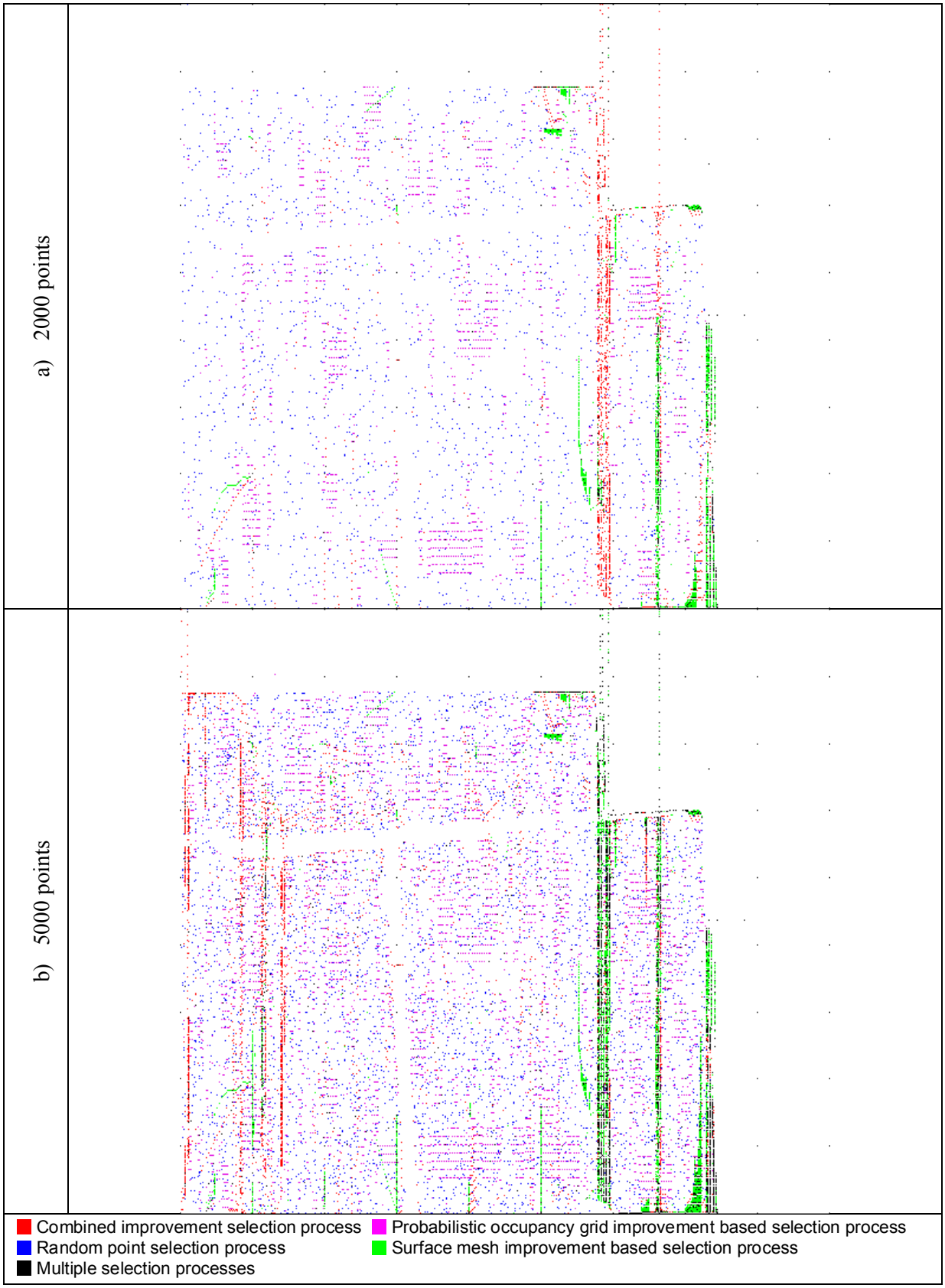


Figure 5.55 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the Jupiter mounted on the linear motorized track on the stacked boxed scene (#18)

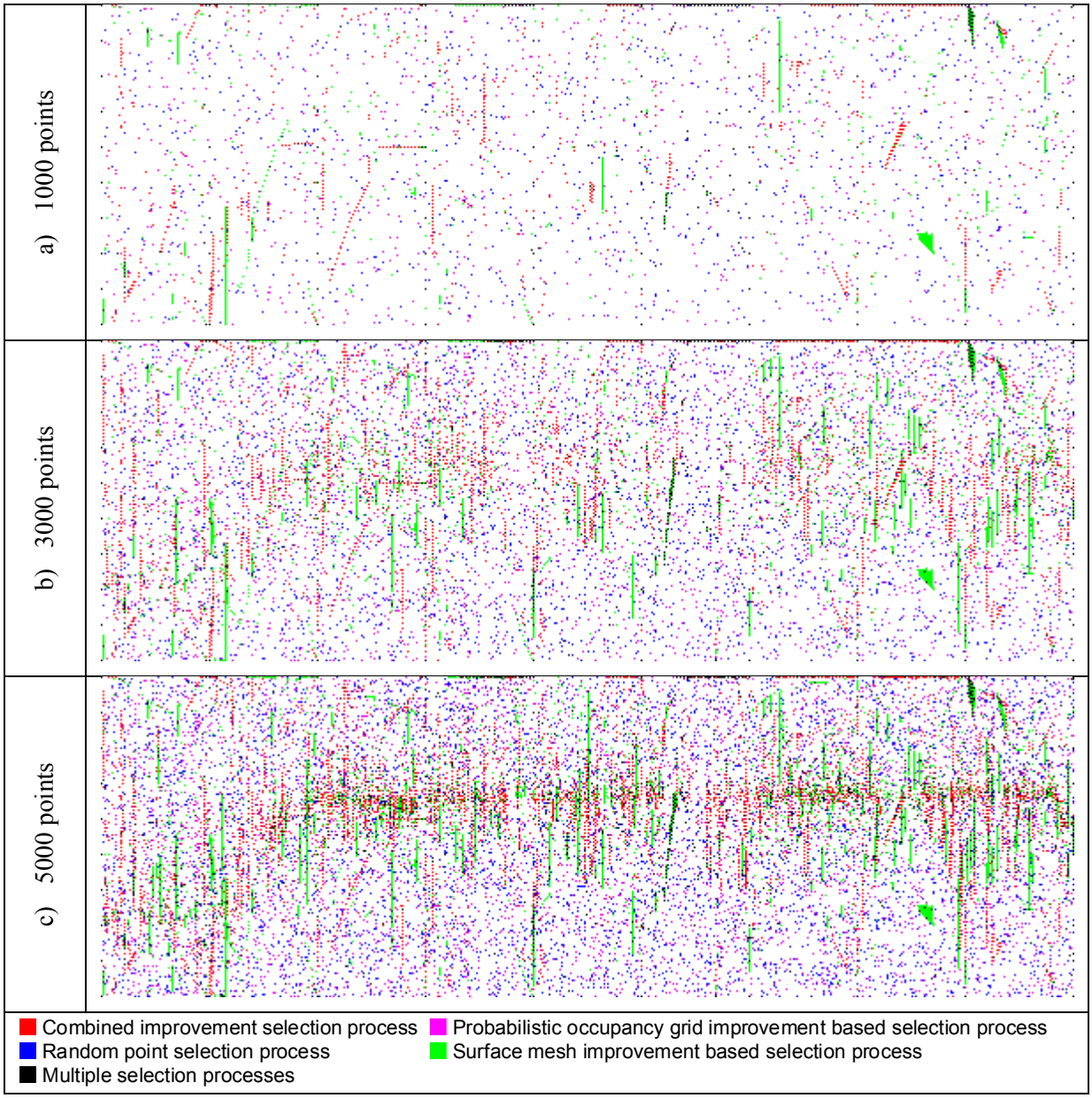


Figure 5.56 - Comparative progression for a) 1000 points, b) 3000 points, and c) 5000 points selectively acquired with the SICK LMS mounted on the linear motorized track on the stacked boxed scene (#18)

5.4.2.3 Summary of Comparative Progression Analysis

The progression of points for the combined improvement selection process, regardless of how many points have been acquired or the sensor employed (as put in evidence in Sections 5.4.2.1, 5.4.2.2, D.7.1, and D.7.2), chooses points that lie near edges and transitions in the scene, while the surface mesh improvement based selection process chooses similar transitions, but acquires each one with more linearly connect points, and thereby provides less coverage than the combined improvement selection process for a given number of selected points. Or in other words, the combined improvement selection process produces a better distribution of points to cover the scene, while at the same time choosing points that accurately represent objects and features within the scene. This verifies the conclusion from the quantitative analysis presented previously in Section 5.4.1. The random point selection process and the probabilistic occupancy grid improvement process choose points that uniformly cover the sensors field of view, although the probabilistic occupancy grid improvement based selection processes did exhibit some patterns, in particular uniformly distributed clusters of points in the case of the ASLS acquisitions, and dotted line acquisitions that were uniformly distributed in the acquisitions of the scenes acquired by the Jupiter mounted on the F3T. The patterns formed in the case of the ASLS acquisition are likely present as a result of the projected pattern forming tight clusters of points, as well as the relatively narrow OPDF variances/region of support used for this sensor. Finally, while the combined improvement selection process did not provide as uniform coverage as did the random point selection and probabilistic occupancy improvement based selection processes, it did provide more accurate representation of features than those selection processes, while at the same time producing a fair distribution of points compared to the surface mesh improvement based selection process.

5.4.3 Data Reduction Analysis

This section provides an analysis of the quality achievable in the estimated surface mesh depth map for a given amount of data reduction, for the random point selection process, the surface mesh improvement based selection process, and the combined improvement selection process. This is accomplished by analysing the aggregate graphs over each scene for the sensors employed in this thesis, for the data savings ratio against PSNR. The data savings ratio is calculated as shown in eq. (5.3), where $DSR(n)$ is the data savings ratio after n points

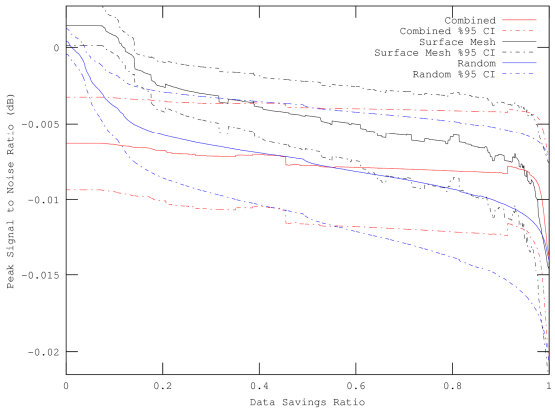
have been acquired out of a total of N possible points. Note that a higher PSNR for a given data savings ratio implies a higher quality representation of the scene, and a higher data savings ratio means that less points are captured.

$$DSR(n) = 1 - \frac{n}{N} \tag{5.3}$$

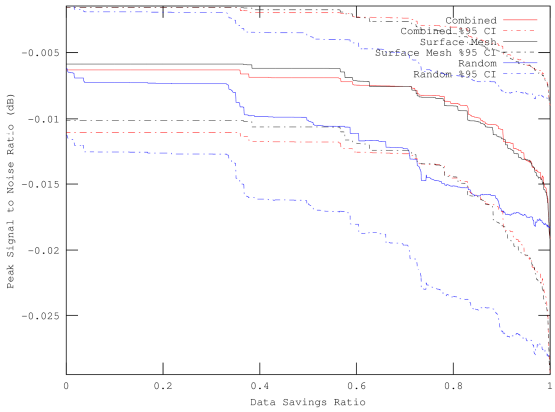
The data savings ratio for the Kinect, Figure 5.57 a), indicates that the combined improvement selection process out performs the other two selection processes for high data savings ratios (greater than about 0.97). This corresponds to approximately 9000 locations selected in the Kinect's field of view for the size of datasets considered in this study, and only a loss of approximately 0.0025 dB in the mean PSNR compared to the complete depth map. The data savings ratio for the ASLS, Figure 5.57 b), indicates that the combined improvement selection process achieves a better PSNR for a data savings ratio greater than about 0.65, which is slightly less (0.0025 dB) than the highest PSNR evident in the plot. Although the total number of locations acquirable by the ASLS, as defined by the OPDF parameters in Table 4.1 from Section 4.1.7, is 1447680, the maximum number of points acquired by the ASLS in the scenes presented in this thesis is 312078, which is the value of *N* used for this graph. This implies that to achieve close to maximum quality, only about 110000 points are required to be acquired. This value is much higher than the values obtainable for the other sensor configurations, as the ASLS has many outliers that cause the PSNR of the estimated depth map to be lower than what would be possible without outliers. The data savings ratio for the Jupiter mounted on the F3T robotic arm, Figure 5.57 c), is larger for the combined improvement selection process for data savings ratios larger than about 0.9, which also corresponds to where the PSNR starts to rapidly decay. A data savings rate of 0.9 for this sensor configuration corresponds to 9200 points out of a total 92160 locations within the field of view, with only a mean 0.0025 dB difference in PSNR from the complete depth map. Figure 5.57 d) illustrates the data savings rate for the three selection processes when choosing points acquired with the Jupiter sensor mounted on the motorized track. The combined improvement selection process performs better at all ratios than do the other selection processes. As evidenced by the lower PSNR with this sensor configuration compared to the other sensor configurations within Figure 5.57, there is more noise present that the combined and surface mesh improvement based selection process try to capture (also

put into evidence in Section 5.4.2), which makes choosing an appropriate data savings ratio goal difficult. A data savings ratio, of 0.95, which corresponds to the region where the surface mesh improvement based selection process starts dropping significantly, results in only 14000 out of a possible 281600 locations within the sensor field of view being acquired, producing an estimated depth map with mean PSNR 0.01 dB less than that of the complete depth map. Figure 5.57 e) presents the mean PSNR for a given data savings ratio possible for a scene acquired with the SICK LMS mounted on the linear motorized track. For this sensor, the combined improvement selection process performs better than the other selection processes for most data savings ratios possible. This graph presents two regions where the PSNR drops, the first right at the start between 0 and 0.2, and the second starting at approximately 0.9. A data savings ratio of 0.9, which corresponds to the acquisition of about 10000 out of 99550 possible locations within the sensor field of view, has a mean PSNR loss from the complete depth map of 0.025 dB. Further gain in the PSNR is not achieved until a data savings ratio of approximately less than 0.2. Finally the results from the Neptec LMS shown in Figure 5.57 f) indicates that for the one scene that the Neptec LMS captured a data saving ratio of 0.2 when guided by either the combined improvement selection process or the surface mesh improvement based selection process yields essentially lossless compression for this scene.

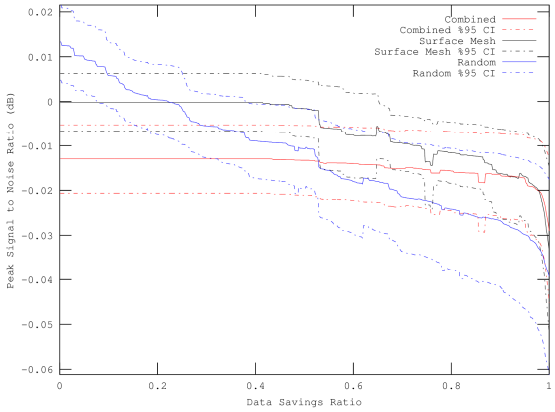
As previously assumed in Chapter 1, there are many redundant points in a range acquisition. For most sensors, using the combined improvement selection process, a large data savings ratio (greater the 0.9) is possible, resulting in only 9000-14000 points being chosen out of all the locations available in the sensor field of view. This corresponds to the results obtained in previous analysis presented in Sections 5.2.1, 5.3.1, and 5.4.1 indicating that the major features are represented with high accuracy after 5000-10000 points have been acquired. For sensors with more outliers and noisy profiles, as put into evidence by Figure 5.57 b) from the ASLS and Figure 5.57 d) from the Jupiter mounted on the track respectively, more points are required to achieve a better PSNR. For the other sensors, the PSNR trend is rather flat until a low data saving rate of approximately 0.2, where the PSNR increases as the data savings rate decreases.



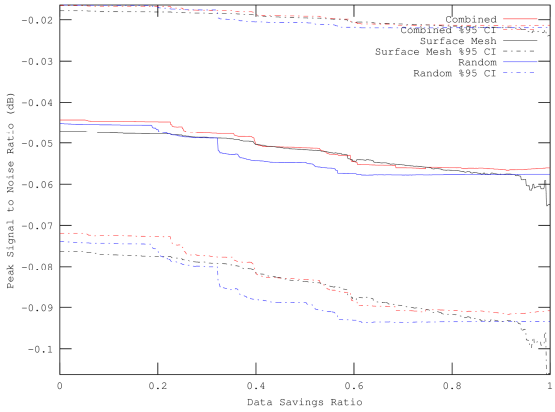
a) Kinect



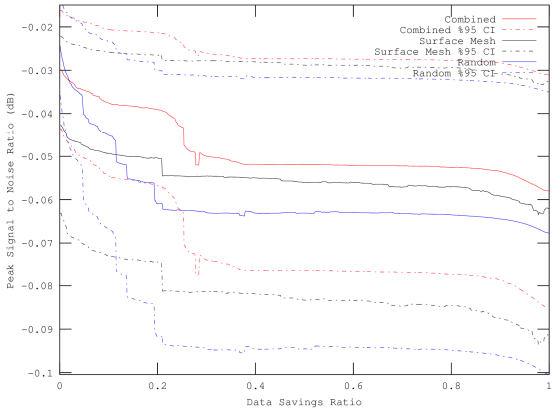
b) ASLS



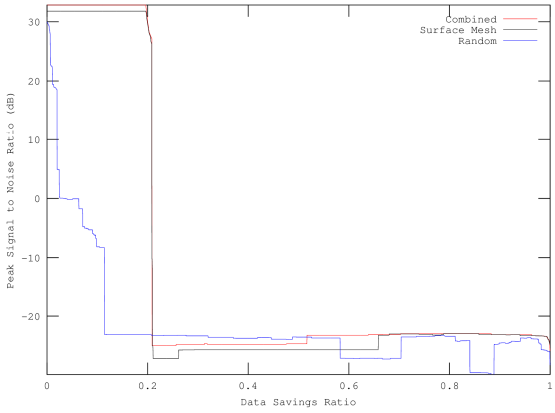
c) Jupiter mounted on F3T



d) Jupiter mounted on track



e) SICK LMS mounted on track



f) Neptec LMS

Figure 5.57 - Data savings ratio vs. PSNR comparison of selection processes for various sensors

5.4.4 Summary of the Evaluation for the Combined Improvement Measure Selection Process

The experimental study conducted in this section puts into evidence that the combined selection process does indeed provide the accuracy demonstrated by the surface mesh improvement based selection process, while ensuring that locations are not acquired adjacent to each other, thereby improving coverage. The analysis consisted of a quantitative study using metrics that measure scene accuracy and coverage, as well as providing an indication of a possible use of the improvement measure to provide information on when to cease acquisition, as a stopping criterion. A study showing the comparative locations chosen by the different techniques was performed to provide visual qualitative evidence of the quantitative performance. Finally a study of the quality (PSNR) given a certain amount of data reduction was performed.

Section 5.4.1 provided an analysis of the performance of the combined improvement selection process against that of the surface mesh improvement based selection process, the probabilistic occupancy grid based selection process, and the random point selection process based on the mean and 95% confidence interval of several metrics for each sensor calculated over several scenes. The surface mesh mean estimated improvement indicated that the proposed combined improvement selection process performed better at accurately representing a scene with fewer points than that obtainable using the proposed surface mesh improvement based selection process alone. The probabilistic occupancy grid mean estimated improvement indicated that the combined improvement selection process provided better scene coverage than that of the surface mesh improvement based selection process, but not as good as that produced by the probabilistic occupancy grid based improvement selection process.

A study of a metric to possibly indicate when to stop acquisition was made using the surface mesh estimated improvement. This is chosen as a possible metric due to it requiring no addition calculation to determine, as it is inherently calculated when producing the surface mesh improvement map. The combined improvement selection process performed similar to the surface mesh improvement based selection process, indicating that it indeed can be used to signify that a scene has been accurately captured.

The fourth metric studied was the PSNR evolution of the estimated depth map from the surface mesh model. This metric detailed that the combined improvement selection process achieved a better PSNR with fewer points acquired in a similar amount of calculation time, when compared against the random point and surface mesh improvement based selection processes.

Section 5.4.2, which provided a comparative analysis detailing the progression of chosen locations for the four selection processes, indicated that the combined selection process does indeed choose points along more features for a given number of points than does the surface mesh improvement metric. Furthermore, the other two methods (the probabilistic occupancy grid improvement based and the random point selection processes) provide better coverage of the field of view of the sensor.

Section 5.4.3 detailed the data savings possible while maintaining certain levels of quality (as measured by the PSNR) when using the combined improvement selection process, compared against the surface mesh improvement based and the random point selection processes. This measure of compression and quality indicated that, for the sensors that produce low levels of noise and outliers, a great amount of data acquisition can be saved while maintaining a high level of quality, namely only 9000-14000 points out of all the points within the entire sensor's field of view need to be acquired, which corresponds well to the observed results of a scene requiring only between 5000-10000 points to accurately capture the major details using the proposed combined improvement selection process, and confirming the assumption made in Chapter 1 that much of a depth image is redundant, and can be effectively estimated by neighbouring locations.

5.5 Discussion on the Experimental Evaluation

The extensive experimental evaluation performed in this chapter puts into evidence the performance of the proposed selective sensing framework that was developed in Chapter 3. Firstly, Section 5.1 provides the experimental procedures that are used in rest of the chapter. Secondly, Section 5.2 puts into evidence, both through quantitative analysis over several scenes and visual confirmation of the progression on the same scenes, that the proposed surface mesh improvement based selection process is able to more accurately capture a scene with fewer points compared to random point selection. Furthermore it was shown that the

surface mesh improvement map provides better guidance on increasing feature accuracy within a scene than does the neural gas method [6,7], and it does this with better scaling of execution time (factor of 2) for a given number of points inserted into the model. Section 5.3 puts into evidence that, both quantitatively and visually, the proposed probabilistic occupancy grid improvement based selection process directs acquisitions to better cover the scene than does random point selection. Finally, the analysis of the proposed combined improvement measure is presented in Section 5.4 and puts into evidence that the combined improvement measure does indeed complementary combine the proposed surface mesh improvement measure, developed in Section 3.1, that focuses on accurately representing features in a scene as verified in Section 5.2, with the proposed probabilistic occupancy grid improvement measure, developed in Section 3.2, that focuses on scene coverage within a sensor's field of view as verified in Section 5.3. This is demonstrated through the quantitative analysis in Section 5.4.1 showing that the combined improvement selection process provides better scene accuracy with fewer points acquired than does the surface mesh improvement based selection process, also indicating that coverage of much more features is performed better using fewer points. This is further verified by observing maps indicating the actual points selected in Section 5.4.2. Section 5.4.3 provides the experimental verification of the assumption that this thesis is based on, that is a selectively guided acquisition can provide high measure of quality using significantly fewer points than a full scan (10% or less). It is demonstrated that this is best achieved, out of the evaluated selection processes, using the proposed combined improvement selection process. Moreover, the proposed techniques were demonstrated, through the wide range of test scenarios examined in this chapter, to perform reliably independent of the sensor technology and scene content.

Chapter 6. Conclusion

6.1 Summary

The proliferation of sensors employed in machine vision applications are a result of decreasing costs and improved quality of these sensors, as well as increased processing capability [1,2]. Unfortunately, due to the increased rate of data production from the ever increasing quantity and resolution of these sensors, managing the resulting datasets in order to obtain meaningful contextual outputs to the end user, be it man or machine, is quickly becoming an important issue that is termed the 'Big Data' Challenge [3]. This thesis concentrates, not on the fusion of many disparate datasets into a contextually coherent database, but on the reduction of the quantity of data acquired by the sensors themselves, namely in the context of reducing the number of locations that need to be captured from within a single point of view of a range sensor. The reduction of the number of range locations needed to represent an arbitrary scene is possible due to surfaces on objects being relatively constant in local regions, thereby allowing the estimation of unmeasured locations within these highly correlated neighbourhoods without formal acquisition of extra measurements. Furthermore, if intelligent techniques are implemented to perform the selection of which points to acquire, further reduction of the quantity of points is possible, while maintaining high quality representation of the scene.

This thesis addresses the single point of view intelligent point selection problem by developing two independent improvement metrics that predict quantitatively the expected improvement in the knowledge of the scene that acquiring any particular location within the sensor's field of view will provide, based upon the already captured knowledge of the same scene. The first proposed improvement metric, based upon surface meshes and the formalism of Kriging, predicts maximum improvement values on locations that increase the accuracy of feature representation within the internal model of the scene. The second proposed improvement metric, based upon probabilistic occupancy grids, predicts maximum improvement values on locations that increase the overall coverage within the sensor's view of the scene. Furthermore, three original point selection processes are developed based upon these improvement metrics; the first uses the surface mesh improvement measure to direct future acquisitions to increase the accurate representation of the scene, the second uses the

probabilistic occupancy grid improvement measure to direct future acquisitions to increase coverage of the scene, and the third utilizes a combination of both improvement measures to accurately capture the scene, while taking into account coverage. Finally, by reducing the number of points needed to represent an arbitrary scene, storage and transmission requirements are reduced, with the additional goal of providing more time to analyse the scene so that actions may be taken sooner.

Chapter 2 provides a review of current methods of both local selective sensing techniques and global multiple viewpoint techniques. In addition, a review of methods that are required to fully implement a data driven selective sensing algorithm are presented. Unlike current techniques for selective sensing, which are limited to cumbersome interruption of a neural network training [6,7], have set deterministic behaviour regardless of underlying data [8], are limited to specific applications [9], or use heuristically sub-optimal pattern switching [16], the techniques proposed and developed in this thesis are based on the stochastic properties of the underlying scene, and formulate direct and original estimates into the amount of potential improvement that locations that are candidates for acquisition make to the final representation of the scene.

Chapter 3 develops the novel methods for estimating the potential improvement that a single point acquisition will have on both the surface mesh representation, as well as the probabilistic occupancy grid representation. By using these improvements to create a pair of improvement maps, original and customized selection processes are introduced and developed in order to use these improvement maps to decide which candidate location will be the next one to acquire. Three such selection processes have been developed in this thesis, one based on the surface mesh improvement measure, in Section 3.3.1, one based upon the probabilistic occupancy grid improvement measure, in Section 3.3.2, and one which combines both improvement measures detailed in Section 3.3.3.

Chapter 4 describes the several different sensors, employing different technologies and configurations that are used to collect real raw 3D datasets over a wide range of scenes that are used in the evaluation of the methods and techniques developed in Chapter 3. Moreover, the derivation of the sensor models and related parameters, as well as details on the data collection for each sensor, are discussed.

Chapter 5 performs an extensive experimental evaluation using the sensors, scenes, and parameters detailed in Chapter 4. This experimental evaluation shows that the proposed surface mesh improvement based selection process outperforms random sampling as quantitatively measured by the surface mesh mean estimated improvement and the surface mesh estimated improvement error. Additionally in Section 5.2, it was shown that the surface mesh improvement based selection process favours accuracy over coverage. The limitation of this selection process, however, is that new acquisition locations tend to be chosen really close to already acquired data in the representation. As Section 5.3 shows, the alternative probabilistic occupancy grid improvement measure, which is also introduced in this thesis, performs well at highlighting large regions that need additional coverage. Section 5.4 studies the combination of both improvement metrics in a single selection process, which yields a decision strategy that outperforms the surface mesh improvement based selection process across virtually all metrics evaluated, and additionally outperforms random sampling in producing an estimated depth map with similar PSNR values, but with fewer points while taking the same amount of time, resulting in superior data compression.

As discussed in Section 5.2, and further verified in Section 5.4, the large scale features, and the locations of the edges are generally visible by the time 5000-10000 points are acquired, and in the majority of cases presented, this occurs faster for the combined improvement measure selection process than the other selections processes. Further analysis made in Section 5.4.3, indicates that the estimated depth map produced from the surface mesh model is able to achieve a model with only slightly lower quality, as measured by PSNR, with a data reduction ratio of at least 0.9, which corresponds to the acquisition of a limited set of only 9000-14000 points when these points are selectively acquired using the proposed combined improvement selection process. This illustrates that data driven selective sensing is a viable alternative to brute force raster acquisition methods, especially for single point acquisition sensors, by avoiding the need for the acquisition of over at least 90% of the locations within a sensor's field of view.

This reduction in the overall number of points needed to be acquired aids in the handling of the 'Big Data' challenge, as less memory and bandwidth need to be used to process the acquired dataset, and since this data reduction is handled at acquisition time, the acquisition

of redundant points is avoided. Furthermore if a surface mesh and/or probabilistic occupancy grid is already being used in the application for modeling the scene, the additional time required to perform the improvement calculations and decide on the location where to scan next is able to provide a net benefit to the acquisition process where slower random access sensors are employed. Moreover, while the surface mesh improvement measure was developed using Cartesian geometry, and the probabilistic occupancy grid improvement measure was developed using spherical geometry, the framework of the improvement measures can be easily changed to accommodate other geometries, such as cylindrical, spherical, Cartesian, and projective coordinate systems through simple parameters replacement. By performing this parameters replacement to allow the use of the improvement measures to be used in other coordinate spaces, the generalization of the techniques is possible such that it can be applied to many different sensors and implemented using their native acquisition geometry, thereby avoiding self-occlusions, as mentioned in Section 2.4.1. This was verified in Section 5.4, which evaluated the algorithmic performance of the combined improvement measure selection process utilizing several different sensors that have differing geometries.

6.2 Contributions

The main original contributions of this research are as follows:

1) Two original methods are developed to measure the estimated improvement that further acquiring specific sets of 3D points/regions would provide to a model of a scene, based upon the already available knowledge of that scene. These two metrics include the surface mesh improvement measure (Section 3.1) that drives future acquisitions to improve accuracy of the scene representation, and the probabilistic occupancy grid improvement measure (Section 3.2) that drives future acquisitions to improve coverage of the scene representation. Furthermore, these improvement metrics are developed taking into account their computational feasibility and existing 3D representation techniques which are widely used in 3D imaging and robotics applications. This enables the possibility of data reuse between different elements in a processing system, thereby limiting the total impact that the additional time required to compute the improvement metrics has on the entire system.

2) As part of the development of the two improvement metrics described above, an original adaptation of the Ordinary Kriging stochastic technique is proposed for estimating depth at unmeasured locations in 3D range images. While the use of Kriging is common in the field of geosciences, it has not widely been applied to machine vision activities, where the benefit of providing both an estimated value and its coinciding variance can aid in a variety of tasks, such as the data-driven selective sensing presented in this thesis.
3) Based on the proposed improvement metrics, further contributions are made through the development of three automated decision methods for automatically and efficiently determining the optimal location for the next acquisition. The first method uses the surface mesh improvement measure (Section 3.3.1). The second method uses the probabilistic occupancy grid improvement measure (Section 3.3.2). The third method uses a selection process that combines the information from both improvement measures in order to balance the demands of accuracy and coverage (Section 3.3.3).
4) An extensive experimental evaluation of the two improvement metrics and the three point selection processes is performed (Chapter 5). This evaluation is entirely based on real raw 3D data acquired using several different sensors over many scenes (Chapter 4), therefore involving implementation of the proposed methods and evaluations on data that inherently involve noise and sensor imperfections. This experimental evaluation demonstrates the independence of the developed techniques to any particular sensor technology, and the robustness of the developed techniques to scene contents. Furthermore, a comparative evaluation is performed against random sampling, and a neural-gas technique [6,7].

This thesis contributes original methods to the field of 3D imaging that support the intelligent and efficient acquisition of arbitrary scenes by developing techniques to estimate the potential improvement to a 3D model that the acquisition of a particular location within a sensor's field of view provides, and then acts on these estimations to measure the depth corresponding to the selected location. These proposed methods can be used in autonomous robotics, search and rescue operations, surveillance systems, 3D segmentation, and remote systems monitoring to reduce the quantity of acquired data required in order to model a scene

from a given point of view. The reduction of the amount of data required for a 3D acquisition of a scene provides a partial solution to the 'Big Data' challenge. Furthermore, the techniques have been demonstrated to be robust against a variety of scenes and sources of noise, and are general enough that they can be employed for sensors of various different underlying technologies and geometries.

Until now, from the work presented in this thesis, and related research activities, the following publications have been made [75,104,105,101,102,90,106]. The segmentation method presented at ROSE 2010 [75] was a result of an initial investigation into producing estimates of where points needed to be added to a 3D scan, but was deemed too complex and unwieldy to be of practical use for a general selective sensing technique. Section 3.2 on the development of the probabilistic occupancy grid based improvement measure was presented at ROSE 2012 [104], and Section 3.1 on the development of the surface mesh based improvement measure was presented at CRV 2013 [105]. [101] was presented at SMC 2013 and details the development of Section 3.1 along with providing a formal comparison of performance of the surface mesh improvement measure against the neural gas method based on data from the Microsoft Kinect. [102] was presented at ACIVS 2013 and details the development of Section 3.1, along with providing a more comprehensive comparison of performance of the surface mesh improvement against the neural gas method based on data from both the Microsoft Kinect and the ASLS as was presented in Section 5.2.4. The final two publications listed here, [90] and [106], report on work performed on the development of range sensing platforms, in margin of the current research, which was beneficial to support the extensive experimental validation of the original concepts developed in this thesis.

6.3 Future Work

There are several extensions that can be developed based upon the work presented in this thesis. The first would be the development of an algorithm to determine when to stop a selective acquisition. As hinted in Section 5.4.1.3, the surface mesh estimated improvement would be the metric of choice to monitor for a stopping condition, potentially using an adaptive filter approach, such a Kalman filter, due to the underlying noise present in this signal.

The consideration of integrating multiple viewpoints in order to fully construct a model of the environment using the selection processes is a promising direction for further investigation. A possible method to perform this, is to project already measured points that are observable from the current viewpoint, but that have been acquired from other viewpoints, into the initial subsampling of the scene. Assuming that the registration between viewpoints is known, this will reduce the computational burden, and redundancies, of acquiring more points in an already known region. Finally, the selection of future viewpoints can be achieved via one of the several NBV algorithms, while fusing the surface mesh and probabilistic occupancy grids using the appropriate data fusion algorithms.

As seen in Chapter 5, the results from applying the selection processes yield results that are desirable in 3D imaging. Firstly, as both improvement measures provide distinct separation of regions, the investigation of their use as effective segmentation algorithms should be performed. Secondly, the evaluation of the combined improvement measure selection process outperformed random sampling in terms of quality of estimated depth map when measured by PSNR versus time as seen in Section 5.4.1.4, and furthermore high data saving ratios are possible that result in only a minimal PSNR penalty as was presented in Section 5.4.3. This suggests the potential of using this method for the compression of surface meshes and depth maps, which should be explored in more detail.

References

[1] Encyclopædia Britannica Online. (2013, May) Moore's law. http. [Online]. http://www.britannica.com/EBchecked/topic/705881/Moores-law

[2] S. Higginbotham. (2011, May 15) Question Everything: A New Processor For Big Data. [Online]. http://gigaom.com/2011/05/15/question-everything-a-new-processor-for-big-data/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:%20OmMalik%20GigaOM:%20Tech/

[3] L. G. Weiss, "Autonomous Robots in the Fog of War," *IEEE Spectrum*, vol. 48, no. 8, pp. 30-34, 56-57, August 2011.

[4] P. McFedries, "The Coming Data Deluge," *IEEE Spectrum*, vol. 48, no. 2, p. 19, February 2011. [Online]. http://spectrum.ieee.org/at-work/innovation/the-coming-data-deluge

[5] R. Washington, K. Golden, J. Bresina, D. E. Smith, C. Anderson, and T. Smith, "Autonomous Rovers for Mars Exploration," in *IEEE Aerospace Conference*, Aspen, CO, 6-13 Mar. 1999, pp. 237-251.

[6] A.-M. Cretu, P. Payeur, and E. M. Petriu, "Selective Vision Sensing with Neural Gas Networks," in *IEEE International Instrumentation and Measurement Technology Conference*, Victoria, BC., 12-15 May 2008, pp. 478-483.

[7] A.-M. Cretu, P. Payeur, and E. M. Petriu, "Selective Range Data Acquisition Driven by Neural Gas Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 8, pp. 2634-2642, August 2009.

[8] C. S. Shih, L. A. Gerhardt, W. C.-C. C. C. Lin, C.-H. Chang, C.-H. Wan, and C.-S. Koong, "Non-Uniform Surface Sampling Techniques for Three-Dimensional Object Inspection," *Optical Engineering*, vol. 47, no. 5, p. 053606, May 2008.

[9] C. Ho and S. Saripalli, "Where Do You Sample? - An Autonomous Underwater Vehicle Story," in *IEEE International Symposium on Robotic and Sensors Environments*, Montreal, QC, 17-18 Sep. 2011, pp. 119-124.

[10] A. Mandow, J. L. Martínez, A. J. Reina, and J. Morales, "Fast Range-Independant Spherical Subsampling of 3D Laser Scanner Points and Data Reduction Performance Evaluation for Scene Registration," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1239-1250, 2010.

[11] A. Torsello, E. Rodolà, and A. Albarelli, "Sampling Relevant Points for Surface Registration," in *IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, Hangzhou, China, May 16-19, 2011, pp. 290-295.

[12] P. J. Besl and N. D. McKay, "A Method for Registraion of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[13] X. Zhu and X. Wu, "Data Acquisition with Active and Impact-Sensitive Instance Selection," in *IEEE International Conference on Tools with Artificial Intelligence*, Boca Raton, FL., 15-17 Nov. 2004, pp. 721-726.

[14] D. Williams, X. Liao, and L. Carin, "Active Data Acquisition with Incomplete Data," Duke University, Dept. of Electrical and Computer Engineering, Durham, NC, Technical Report, Oct. 2005.

[15] M. Sheehan, A. Harrison, and P. Newman, "Self-Calibration for a 3D Laser," *The International Journal of Robotics Research*, vol. 5, no. 31, pp. 675-687, 2012.

[16] C. English, G. Okouneva, P. Saint-Cyr, A. Choudhuri, and T. Luu, "Real-Time Dynamic Pose Estimation Systems in Space: Lessons Learned for System Design and Performance Evaluation," *International Journal of Intelligent Control and Systems*, vol. 16, no. 2, pp. 79-96, June 2011.

[17] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling Visual Attention via Selective Tuning," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 507-545, Oct. 1995.

[18] N. D. B. Bruce and J. K. Tsotsos, "Saliency, Attention, and Visual Search: An Information Theoretic Approach," *Journal of Vision*, vol. 9, no. 3, pp. 1-24 (Article 5), March 2009.

[19] C. I. Connolly, "The Determination of Next Best Views," in *IEEE International Conference on Robotics and Automation*, vol. 2, St. Louis, MO, 25-28 Mar. 1985, pp. 432-435.

[20] J. Maver and R. Bajcsy, "Occlusions as a Guide for Planning the Next View," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, pp. 417-433, May 1993.

[21] V. Sequeira, J. G. M. Goncalves, and M. I. Ribeiro, "Active View Selection for Efficient 3D Scene Reconstruction," in *IEEE International Conference on Pattern Recognition*, vol. 1, Vienna, Austria, 25-29 Aug. 1996, pp. 815-819.

[22] R. Pito, "A Solution to the Next Best View Problem for Automated Surface Acquisition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1016-1030, October 1999.

[23] H. Zha, K. Morooka, and T. Hasegawa, "Active Modeling of 3-D Objects: Planning on the Next Best Pose (NBP) for Acquiring Range Images," in *IEEE International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, Ottawa, ON, 12-15 May 1997, pp. 68-75.

[24] K. Morooka, H. Zha, and T. Hasegawa, "Computations on a Spherical View Space for Efficient Planning of Viewpoints in 3-D Object Modeling," in *IEEE International Conference on 3-D Digital Imaging and Modeling*, Ottawa, ON., 4-8 Oct. 1999, pp. 138-147.

[25] M. K. Reed, P. K. Allen, and I. Stamos, "3-D Modeling from Range Imagery: An Incremental Method with a Planning Component," in *IEEE International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, Ottawa, ON., 12-15 May 1997, pp. 76-83.

[26] L. M. Wong, C. Dumont, and M. A. Abidi, "Next Best View System in a 3-D Object Modeling Task," in *IEEE International Symposium on Computational Intelligence on Robotics and Automation*, Monterey, CA., 8-9 Nov. 1999, pp. 306-311.

[27] K. Klein and V. Sequeira, "The View-Cube: An Efficient Method of View Planning for 3D Modelling from Range Data," in *IEEE Workshop on Applications of Computer Vision*, Palm Springs, CA., 4-6 Dec. 2000, pp. 186-191.

[28] D. K. MacKinnon, "Adaptive Laser Range Scanning," Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, PhD Thesis, 2008.

[29] D. MacKinnon, V. Aitken, F. Blais, and M. Picard, "Adaptive Laser Range Scanning," in *IEEE International Workshop on Robotic and Sensors Environments*, Ottawa, ON., 12-13 Oct. 2007, pp. 114-119.

[30] D. MacKinnon, V. Aitken, and F. Blais, "Adaptive Laser Range Scanning," in *American Control Conference*, 2008, pp. 3857-3862.

[31] W. R. Scott, G. Roth, and J.-F. Rivest, "View Planning for Automated Three-Dimensional Object Reconstruction and Inspection," *ACM Computing Surveys*, vol. 35, no. 1, pp. 64-96, March 2003.

[32] H. González-Banos and J.-C. Latombe, "A Randomized Art-Gallery Algorithm for Sensor Placement," in *Proceedings of the Symposium on Computational Geometry*, New York, NY, USA, 2001, pp. 232-240.

[33] C. H. J. Beaven, *The Approximations Handbook - The Art of Nearly*. Wilmslow, England: Sigma Press, 1989.

[34] L. Komzsik, *Approximation Techniques for Engineers*. Boca Raton, Florida, USA: CRC Press, 2007.

[35] A. M. Herzberg and D. R. Cox, "Some Optimal Designs for Interpolation and Extrapolation," *Biometrika*, vol. 59, no. 3, pp. 551-561, Dec. 1972.

[36] T. J. Rivlin, *An Introduction to the Approximation of Functions*. Waltham, Massachusetts, USA: Blaisdell Publishing Company, 1969.

[37] T. N. E. Greville, Ed., *Theory and Applications of Spline Functions*. Madison, Wisconson, USA: Academic Press, 1968.

[38] K. L. Rasmussen and P. V. Sharma, "Bicubic Spine Interpolation: A Quantitative Test of Accuracy and Efficiency," *Geophysical Prospecting*, vol. 27, no. 2, pp. 394-408, 1979.

[39] S. Haykin, *Neural Neworks A Comprehensive Foundation*, 2nd ed. Upper-Saddle River, New Jersey, USA: Prentice-Hall Inc., 1999.

[40] F. M. Ham and I. Kostanic, *Principles of Neurocomputing for Science & Engineering*. New York, New York, USA: McGraw-Hill, 2001.

[41] L. Fu, *Neural Networks in Computer Intelligence*. New York, New York, USA: McGraw Hill, Inc., 1994.

[42] C. K. I. Williams, "Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond," Neural Computing Research Group, Aston University, Birmingham, U.K., Technical Report, 1997.

[43] G. Bohling, "Kriging," 2005. [Online]. people.ku/edu/~gbohling/cpe940/Kriging.pdf

[44] P. Goovaerts, *Geostatistics for Natural Resources Evaluation*. New York, New York, USA: Oxford University Press, 1997.

[45] I. Clark, *Practical Geostatistics*. Alloa, Central Scotland, July 2001. [Online]. http://www.kriging.com/PG1979/PG1979_pdf.html

[46] Grains Research & Development Corporation, "Making Yield Maps: A guide for the Australian Grains Industry,". [Online]. www.usyd.edu.au/agriculture/acpa/documents/YieldMapping.pdf

[47] L. E. Dunlap and J. M. Spinazola, "Interpolating Water-Table Altitudes in West-Central Kansas Using Kriging Techniques," U.S Department of the Interior, United States Geological Survey, Washington DC, Water-Supply Paper, 1984.

[48] B. Grinstead, A. Koschan, and M. A. Abidi, "Geometry Refinement of 3D Surfaces Using Kriging," in *IEEE Third International Symposium on 3D Data Processing, Visualization, and Transmission*, Chapel Hill, NC, 14-16 June 2006, pp. 719-726.

[49] T. C. Haas, "Kriging and Automated Variogram Modeling Within a Moving Window," *Atmospheric Environment*, vol. 24a, no. 7, pp. 1759-1769, 1990.

[50] D. Nordman, "An Empirical Likelihood Method for Variogram Estimation," in *International Workshop on Spatio-Temporal Modelling*, Pamplona, Spain, 27-29 Sept. 2006.

[51] E. Pardo-Igúzquiza, "MLREML4: A Program for the Inference of the Power Variogram Model by Maximum Likelihood and Restricted Maximum Likelihood," *Computers & Geosciences*, vol. 24, no. 6, pp. 537-543, 1998.

[52] U. N. Bhat and G. K. Miller, *Elements of Applied Stochastic Processes*, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons Inc., 2002.

[53] J. Pineau, G. Gordon, and S. Thrun, "Point-Based Value Iteration: An Anytime Algorithm for POMDPs," in *International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 9-15 Aug. 2003, pp. 1025-1032.

[54] A. R. Cassandra. (1999, Jan.) Tony's POMDP Page. [Online]. www.cs.brown.edu/research/ai/pomdp/index.html

[55] B. Givan and R. Parr. (2001, Nov.) An Introduction to Markov Decision Processes. [Online]. http://www.cs.rice.edu/~vardi/dag01/givan1.pdf

[56] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien, "Acting Under Uncertainty: Discrete Bayesian Models for Mobile Robot Navigation," in *IEEE International Conference on Intelligent Robots and Systems*, Osaka, Japan, 4-8 Nov. 1996, pp. 963-972.

[57] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall, Inc, 1998.

[58] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, "Delaunay Triangulations: Height Interpolation," in *Computational Geometry: Algorithms and Applications Third Edition*.: Springer-Verlag, 2008, ch. 9, pp. 191-218.

[59] S. Fomel. (2000, Dec.) Incremental Delaunay Triangulation and Related Problems. [Online]. http://sepwww.stanford.edu/oldsep/sergey/sepsergey/fmeiko/paper_html/node9.html

[60] B. Mederos, L. Velho, and L. H. de Figueiredo, "Moving Least Squares Multiresolution Surface Approximation," in *Brazilian Symposium on Computer Graphics and Image*, São Carlos, Brazil, 12-15 Oct. 2003, pp. 19-26.

[61] J. Wang, M. M. Oliveira, and A. E. Kaufman, "Reconstructing Manifold and Non-Manifold Surfaces from Point Clouds," in *IEEE Conference on Visualization*, Minneapolis, MN, 23-28 Oct. 2005, pp. 415-422.

[62] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Close-range Scene Segmentation and Reconstruction of 3D Point Cloud Maps for Mobile Manipulation in Domestic Environments," in *IEEE International Conference on Intelligent Robots and Systems*, St. Louis, MO. USA, 10-15 Oct. 2009, pp. 1-6.

[63] O. Schall, A. Belyaev, and H.-P. Seidel, "Robust Filtering of Noisy Scattered Point Data," in *Eurographics/IEEE VGTC Symposium on Point-Based Graphics*, Stony Brook, NY, 21-22 June 2005, pp. 71-77.

[64] C. I. Connolly, "Cumulative Generation of Octree Models from Range Data," in *IEEE International Conference on Robotics and Automation*, Atlanta, GA, 13-15 Mar. 1984, pp. 25-32.

[65] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer*, vol. 22, no. 6, pp. 46-57, June 1989.

[66] A. Elfes, "Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation," Carnegie Mellon University, Pittsburgh, PA, PhD Thesis, 1989.

[67] P. Payeur, P. Hébert, D. Laurendeau, and C. M. Gosselin, "Probabilistic Octree Modeling of a 3-D Dynamic Environment," in *International Conference on Robotics and Automation*, Albuquerque, NM, 20-25 Apr. 1997, pp. 1289-1296.

[68] M. Pauly, R. Keiser, and M. Gross, "Multi-scale Feature Extraction on Point-Sampled Surfaces," *Computer Graphics Forum*, vol. 22, no. 3, pp. 281-289, Sept. 2003.

[69] H. Woo, E. Kang, S. Wang, and K. H. Lee, "A New Segmentation Method for Point Cloud Data," *International Journal of Machine Tools & Manufacture*, vol. 42, no. 2, pp. 167-178, Jan. 2002.

[70] A. Yogeswaran and P. Payeur, "Features Extraction from Point Clouds for Automated Detection of Deformations on Automotive Body Parts," in *Proceedings of the IEEE International Workshop on Robotic and Sensors Environments*, Lecco, Italy, 6-7 Nov. 2009, pp. 122-127.

[71] A. Yogeswaran and P. Payeur, "3D Surface Analysis for Automated Detection of Deformations on Automotive Body Panels," in *New Advances in Vehicular Technology and Automotive Engineering*, J. P. Carmo and J. E. Ribeiro, Eds. Rijeka, Croatia: InTech, Aug. 2012, ch. 12, pp. 303-332.

[72] A. Hubeli and M. Gross, "Multiresolution Feature Extraction from Unstructured Meshes," in *IEEE Conference on Visualization*, San Diego, CA., 21-26 Oct. 2001, pp. 287-294.

[73] Y. Lee, S. Park, Y. Jun, and W. C. Choi, "A Robust Approach to Edge Detection of Scanned Point Data," *International Journal of Advanced Manufacturing Technology*, vol. 23, no. 3, pp. 263-271, 2004.

[74] B. Kim and P. Tsiotras, "Image Segmentation on Cell-Center Sampled Quadtree and Octree Grids," in *SPIE Wavelet Applications in Industrial Processing VI*, San Jose, CA, USA, 18-22 Jan. 2009.

[75] P. Curtis and P. Payeur, "A Method to Segment a 3D Surface Point Cloud for Selective Sensing in Robotic Exploration," in *IEEE International Workshop on Robotic and Sensors Environments*, Phoenix, AZ, 15-16 Oct. 2010, pp. 152-157.

[76] A. Bovik, Ed., *Handbook of Image & Video Processing*, 2nd ed. Boston, MA, USA: Elsevier Academic Press, 2005.

[77] S. Haykin, *Communications Systems*, 4th ed. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[78] P. Alliez and C. Gotsman, "Recent Advances in Compression of 3D Meshes," in *Proceedings of the Symposium on Multiresolution in Geometric Modeling*, 2003, pp. 3-26.

[79] R. Krishnamurthy, B.-B. Chai, H. Tao, and S. Sethuraman, "Compression and Transmission of Depth Maps for Image-Based Rendering," in *IEEE International Conference on Image Processing*, Thessaloniki, Greece, 7-10 Oct. 2001, pp. 828-831.

[80] M. Sarkis and K. Diepold, "Depth Map Compression via Compressed Sensing," in *IEEE International Conference on Image Processing*, Cairo, Egypt, Nov 7-12, 2009, pp. 737-740.

[81] J. R. Armstrong, "Design of a graphic generator for remote terminal application," *IEEE Transactions on Computers*, vol. C-22, no. 5, pp. 464-468, May 1973.

[82] Microsoft. Kinect for Windows Sensor Components and Specifications. [Online]. http://msdn.microsoft.com/en-us/library/jj131033(d=printer).aspx

[83] B. Klug. (2010, Sep.) AnandTech - Microsoft Kinect: The AnandTech Review. [Online]. http://www.anandtech.com/show/4057/microsoft-kinect-the-anandtech-review/2

[84] PrimeSense. PrimeSense 3D Sensor Data Sheet. [Online]. http://primesense.com/en/press-room/resources/file/4-primesense-3d-sensor-data-sheet

[85] DotNetNuke Corp. (2012, Oct.) OpenNI -> OpenNI > Home. [Online]. http://openni.org/

[86] K. Khoshelham, "Accuracy Analysis of Kinect Depth Data," in *ISPRS Workshop Laser Scanning 2011*, Calgary, AB., 2011.

[87] R. Macknojia, A. Chávez-Aragón, P. Payeur, and R. Laganière, "Experimental Characterization of Two Generations of Kinect's Depth Sensors," in *IEEE International Symposium on Robotic and Sensors Environments*, Magdeburg, Germany, 16-18 Nov. 2012, pp. 150-155.

[88] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, Second Edition ed. San Diego, CA, USA: Harcourt Academic Press, 2000.

[89] A. Boyer, "Adaptive Structured Light Imaging for 3D Reconstruction and Autonomous Robotic Exploration," University Of Ottawa, Ottawa, ON, MASc Thesis, 2009.

[90] A. Boyer, P. Curtis, and P. Payeur, "3D Modeling from Multiple Views with Integrated Registration and Data Fusion," in *Canadian Conference on Computer and Robot Vision*, Kelowna, BC, 25-27 May 2009, pp. 252-259.

[91] D. Desjardins, "Structured Lighting Stereoscopy with Marching Pseudo-Random Patterns," University of Ottawa, Ottawa, ON, MASc Thesis, 2008.

[92] Willow Garage. (2001, 01 July) Camera Calibration and 3D Reconstruction - OpenCV 2.4.6.0 Documentation. [Online]. http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

[93] P. Curtis, "Frequency-Domain Range Data Registration for 3-D Space Modeling in Robotic Applications," University of Ottawa, Ottawa, ON, MASc Thesis, 2006.

[94] P. Curtis and P. Payeur, "An Integrated Robotic Laser Range Sensing System for Automatic Mapping of Wide Workspaces," in *Canadian Conference on Electrical and Computer Engineering*, Niagra Falls, ON, 2-5 May 2004, pp. 1135-1138.

[95] M. Rioux, "Laser Rangefinder Based on Synchronized Scanning," *Applied Optics*, vol. 23, no. 21, pp. 3837-3844, Nov. 1985.

[96] Servo-Robot Inc., *Jupiter 3-D Laser Vision Camera Installation and Operation Manual*. St-Bruno, QC: Servo-Robot Inc., 1996.

[97] CRS Robotics Corporation, *F3 Robot System User Guide*, 001st ed. Burlington, ON, Canada: CRS Robotics Corporation, 2000.

[98] SICK AG. (2012, Dec.) Technical Description LMS200/211/221/291 Laser Measurement Systems. PDF. [Online]. https://www.mysick.com/saqqara/im0012759.pdf

[99] Neptec Technologies Corp. (2012, Nov) Neptec Laser Metrology System (LMS). PDF. [Online]. http://www.neptectechnologies.com/Media/brochures/NTC-Laser-Metrology-System(LMS).pdf

[100] Visual Computing Lab, ISTI-CNR. (2013, 01 July) MeshLab. [Online]. http://meshlab.sourceforge.net/

[101] P. Payeur, P. Curtis, and A.-M. Cretu, "Computational Methods for Selective Acquisition of Depth Measurements in Machine Perception," in *IEEE International Conference on Systems, Man, and Cybernetics*, Manchester, UK, 13-16 Oct. 2013, pp. 876-881.

[102] P. Payeur, P. Curtis, and A.-M. Cretu, "Computational Methods for Selective Acquisition of Depth Measurements: an Experimental Evaluation," in *Advanced Concepts for Intelligent Vision Systems, Lecture Notes in Computer Science 8192*, Poznan, Poland, 28-31 Oct. 2013, pp. 389-401.

[103] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. New York, USA: McGraw Hill, 2002.

[104] P. Curtis and P. Payeur, "Estimating Optimal Regions for Improvement in Range Acquisition from a Single Point of View," in *IEEE International Symposium on Robotic and Sensors Environments*, Magdeburg, Germany, 16-18 Nov. 2012, pp. 198-203.

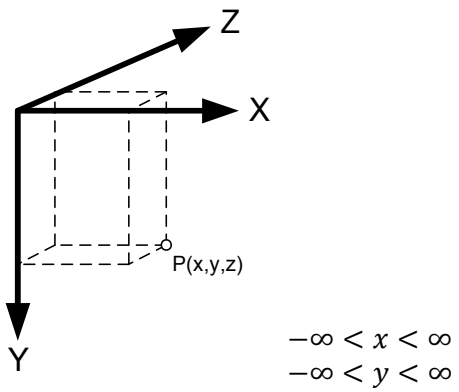
[105] P. Curtis and P. Payeur, "A Method for Dynamic Selection of Optimal Depth Measurements Acquisition with Random Access Range Sensors," in *IEEE International Conference on Computer and Robot Vision*, Regina, SK, Canada, 28-31 May, 2013, pp. 311-318.

[106] P. Curtis, C. S. Yang, and P. Payeur, "An Integrated Robotic Multi-Modal Range Sensing System," in *IEEE International Instrumentation and Measurement Technology Conference*, Ottawa, ON, 17-19 May 2005, pp. 1991-1996.

[107] B. Friedland, *Control System Design*. Mineola, New York, USA: Dover Publications, Inc., 2005.

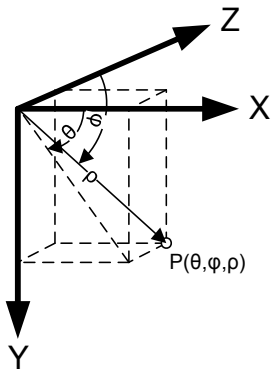
Appendix A. Coordinate Transformations

The convention for the coordinate systems used in this thesis are illustrated in Figure A.1, with a) showing the familiar Cartesian coordinate system, b) showing the Spherical coordinate system, c) showing the Cylindrical coordinate system, and d) showing the Angular Projection coordinate system, all in relation to the Cartesian coordinate system. Eq. (A.1) defines the transformations from Cartesian coordinates (x, y, z) to spherical coordinates (θ, φ, ρ) and back again, eq. (A.2) defines the transformations from Cartesian coordinates (x, y, z) to cylindrical coordinates (x, θ, ρ) and back again, and finally eq. (A.3) defines the transformations from Cartesian coordinates (x, y, z) to angular projection coordinates (θ, φ, ρ) and back again.



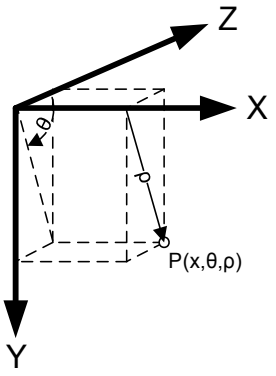
$$-\infty < x < \infty$$
$$-\infty < y < \infty$$
$$-\infty < z < \infty$$

a) Cartesian Coordinates



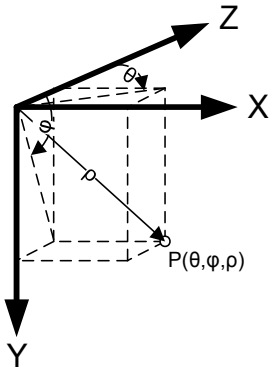
$$-\pi < \theta \leq \pi$$
$$0 \leq \varphi \leq \pi$$
$$0 \leq \rho < \infty$$

b) Spherical Coordinates



$$-\infty < x < \infty$$
$$-\pi < \theta \leq \pi$$
$$0 \leq \rho < \infty$$

c) Cylindrical Coordinates



$$-\pi < \theta \leq \pi$$
$$-\pi < \varphi \leq \pi$$
$$0 \leq \rho < \infty$$

d) Angular Projection Coordinates

Figure A.1 - Diagram illustrating coordinate transformations in relation to Cartesian coordinates

$$\begin{matrix} \theta = atan(y/x) \\ \varphi = acos(z/\rho) \\ \rho = \sqrt{x^2 + y^2 + z^2} \end{matrix} \quad \longleftrightarrow \quad \begin{matrix} x = \rho \cdot sin(\varphi) \cdot cos(\theta) \\ y = \rho \cdot sin(\varphi) \cdot sin(\theta) \\ z = \rho \cdot cos(\varphi) \end{matrix} \tag{A.1}$$

$$\begin{matrix} x = x \\ \theta = atan(y/z) \\ \rho = \sqrt{y^2 + z^2} \end{matrix} \quad \longleftrightarrow \quad \begin{matrix} x = x \\ y = \rho \cdot sin(\theta) \\ z = \rho \cdot cos(\theta) \end{matrix} \tag{A.2}$$

$$\begin{matrix} \theta = atan(x/z) \\ \varphi = atan(y/z) \\ \rho = \sqrt{x^2 + y^2 + z^2} \end{matrix} \quad \longleftrightarrow \quad \begin{matrix} x = z \cdot tan(\theta) \\ y = z \cdot tan(\varphi) \\ z = \dfrac{\rho}{\sqrt{1 + tan^2(\theta) + tan^2(\varphi)}} \end{matrix} \tag{A.3}$$

Note that spherical coordinates are symmetric along θ and φ, such that adding or subtracting π to θ results in a sign change in φ ($\theta + \pi \rightarrow -\varphi$). Using this relationship, it is possible to map the Z-Cartesian axis to correspond to the center of the scene, which is performed in this thesis for the case when the sensor is the Jupiter laser line scanner mounted on the F3T robot.

Appendix B. Block Matrix Inversion

Block matrix inversion, also known as partitioned matrix inversion, is a method to invert a partitioned matrix in terms of its submatrices [107]. Given a square matrix, $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A is an invertible $i \times k$ matrix, B is a $i \times (m-k)$, C is a $(m-i) \times k$ matrix, D is a $(m-i) \times (m-k)$ matrix, $1 \leq i < m$, and $1 \leq k < m$, then its inverse is defined as in eq. (B.1).

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix} \tag{B.1}$$

Appendix C. Datasets

This appendix describes all the datasets acquired for use in this thesis. Each section of this appendix describes a different scene that is used for the evaluation of this thesis. Each section describes the scene, as well as providing a figure that illustrates an RGB image of the scene, in addition to a rendering of the raw 3D point clouds acquired from each sensor (as described in Section 4.1) that was used to capture the scene. The description of each point cloud rendering also contains the number of points that are contained in the corresponding point cloud. Furthermore, the renderings of the 3D point clouds from the Kinect and ASLS have RGB values, in addition to position, and hence they have been rendered in colour. The point clouds from the other sensors have been rendered using only white, as they do not produce any colour information.

C.1 Scene 1: Pillars in a Parking Garage

This scene, shown in Figure C.1, contains a couple of pillars from within an underground parking garage. It was acquired with only the Microsoft Kinect. This scene is interesting due to the varying depth of the pillars, and the fact that the pillars are cylindrical, which can be used to evaluate the performance on curved surfaces.

a) RGB image of the scene

b) Raw 3D point cloud of the scene as captured by the Kinect (254060 points)

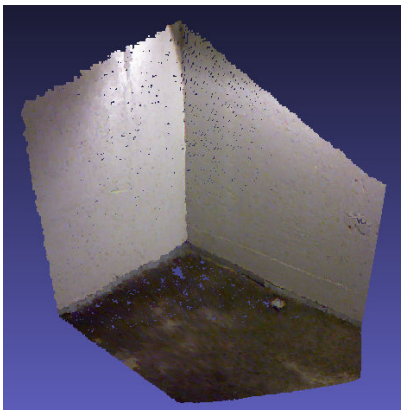
Figure C.1 - Scene 1: pillars in a parking garage

C.2 Scene 2: Corner in a Parking Garage

This scene, shown in Figure C.2, is of an interior 90° corner in a parking garage, and was only acquired with the Kinect. Although it is a relatively simple scene, it is interesting as the main features in the image are the locations where the planar surfaces (the two walls and the

ground) intersect each other, hence the performance of this feature type can be evaluated in isolation with few other features being involved.

a) RGB image of the scene



b) Raw 3D point cloud of the scene as captured by the Kinect (307200 points)

Figure C.2 - Scene 2: corner in a parking garage

C.3 Scene 3: Cluttered Corner in a Parking Garage

This scene, Figure C.3, was only acquired with the Kinect and contains the same interior corner of a parking garage as shown in Figure C.2, but with additional objects (3 traffic cones and a filled recycling box). This provides a scene that contains the 90° interior corner, in addition to other objects at various depth.

a) RGB image of the scene

b) Raw 3D point cloud of the scene as captured by the Kinect (307200 points)

Figure C.3 - Scene 3: cluttered corner in a parking garage

C.4 Scene 4: Car and Walls in a Parking Garage

This scene, as shown in Figure C.4, consists of a car parked in a parking garage beside a wall, and was only acquired with the Kinect. The differing densities of the acquisition due to the unacquirable surfaces, such as the window, the reflector, and oblique occluded surfaces,

make this scene interesting for evaluating the performance of the algorithms developed in Chapter 3.

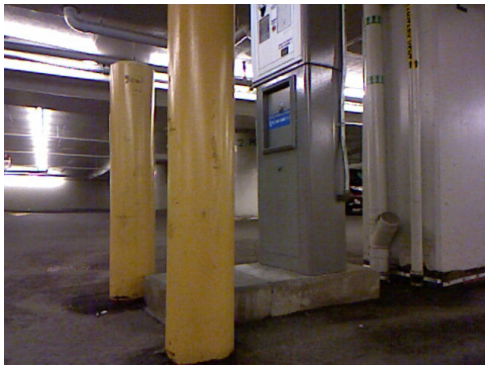
a) RGB image of the scene

b) Raw 3D point cloud of the scene as captured by the Kinect (279299 points)

Figure C.4 - Scene 4: car and walls in a parking garage

C.5 Scene 5: Pay Machine and Pillars in a Parking Garage

This scene, as shown in Figure C.5, consists of a pay machine and some pillars in a parking garage, and was only acquired by the Kinect. This scene is interesting as it contains surfaces that are out of range of the Kinect, in addition to both the cylindrical pillars, and the rectilinear pay machine, which are at different depths.



a) RGB image of the scene

b) Raw 3D point cloud of the scene as captured by the Kinect (259676 points)

Figure C.5 - Scene 5: pay machine and pillars in a parking garage

C.6 Scene 6: Cluttered Scene in a Lab

This scene, shown in Figure C.6, consists of several overlapping objects at various depths in order to create a cluttered scene in a lab. This scene, which was acquired only with the Kinect, has several interesting properties. Firstly, the black fender was not significantly acquired by the Kinect, which results in a gap within the acquired data. Secondly the small

concave features of the solid wooden pallet are interesting, as they may be missed by an initial sampling, thereby providing details that would require some additional coverage of the scene in addition to determine their locations.

a) RGB image of the scene

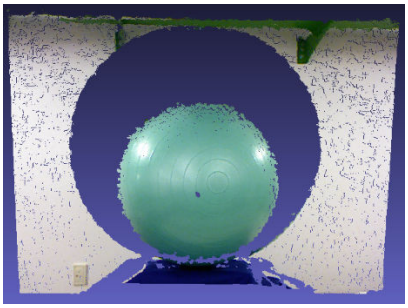
b) Raw 3D point cloud of the scene as captured by the Kinect (277753 points)

Figure C.6 - Scene 6: cluttered scene in a lab

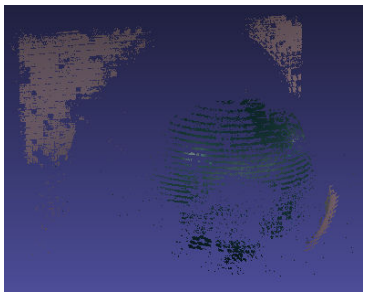
C.7 Scene 7: Exercise Ball

This scene of an exercise ball, shown in Figure C.7, was acquired with the Kinect, ASLS, Jupiter mounted on the F3T robot arm, and the SICK LMS mounted on the linear motorized track. The spherical shape of the exercise ball makes it an interesting scene to capture, and use for the evaluation of the algorithms developed in Chapter 3. Furthermore the surface of the ball is a green rubber, which is difficult to capture using red lasers, hence the poor acquisition using the Jupiter.

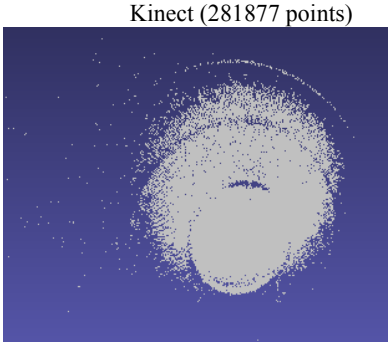
a) RGB image of the scene



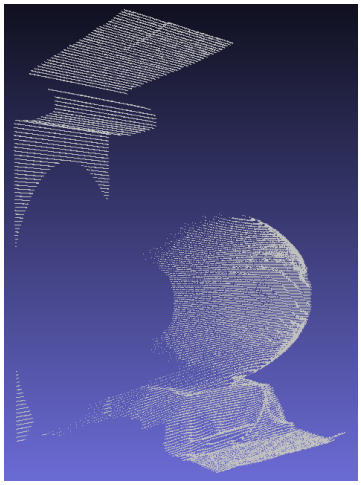
b) Raw 3D point cloud of the scene as captured by the Kinect (281877 points)



c) Raw 3D point cloud of the scene as captured by the ASLS (77555 points)



d) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (87552 points)



e) Raw 3D point cloud of the scene as captured by the SICK/Track (99550 points)

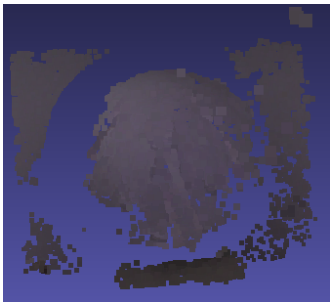
Figure C.7 - Scene 7: exercise ball

C.8 Scene 8: Exercise Ball Cover with a Sheet

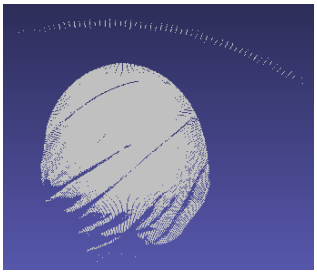
This scene, shown in Figure C.8, is the same exercise ball shown in Figure C.7, but it has a white sheet covering it, thereby making it better to acquire for the red laser of the Jupiter sensor. The Kinect, ASLS, Jupiter mounted on the F3T robotic arm, Jupiter mounted on the linearized track, and the SICK LMS mounted on the motorized track acquired the scene. The track based sensors performed two acquisitions since both sensors capture ranged data at the same time, but are spaced apart from each other, and as such are unable to have a full view of the scene simultaneously, hence the entire track is shifted to allow each sensor an opportunity to have complete coverage of the scene. The folds on the sheet create small features that are of additional interest for the evaluation of the proposed algorithms from Chapter 3.

a) RGB image of the scene

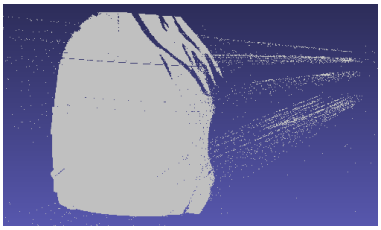
b) Raw 3D point cloud of the scene as captured by the Kinect (286942 points)



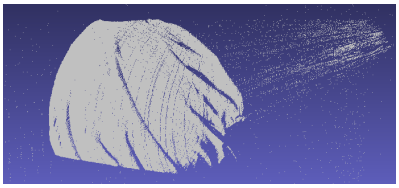
c) Raw 3D point cloud of the scene as captured by the ASLS (36337 points)



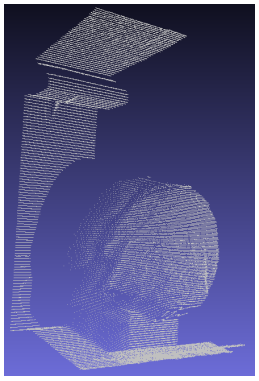
d) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (69100 points)



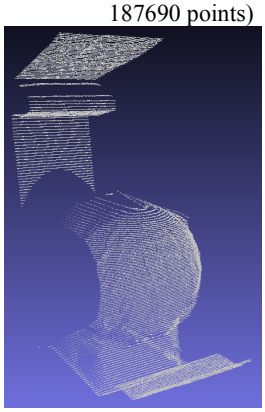
e) Raw 3D point cloud of the scene as captured by the Jupiter/Track (left side, 187690 points)



f) Raw 3D point cloud of the scene as captured by the Jupiter/Track (right side, 152853 points)



g) Raw 3D point cloud of the scene as captured by the SICK/Track (left side, 99550 points)



h) Raw 3D point cloud of the scene as captured by the SICK/Track (right side, 99550 points)

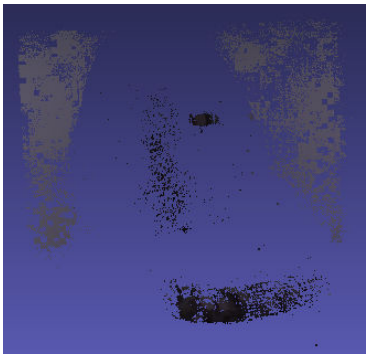
Figure C.8 - Scene 8: exercise ball cover with a sheet

C.9 Scene 9: Textured Chair

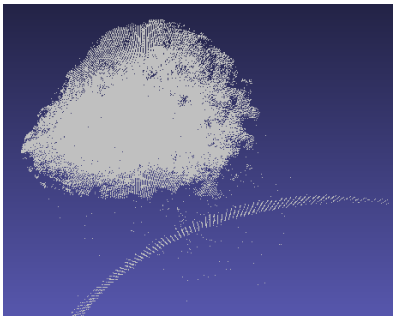
This scene, as shown in Figure C.9, contains a textured chair, and was acquired with the Kinect, ASLS, the Jupiter mounted on the F3T robotic arm, and the SICK mounted on the motorized track. The chair contains concave curved surfaces (the back and the seat) that are interesting for the evaluation of the proposed techniques in Chapter 3.

a) RGB image of the scene

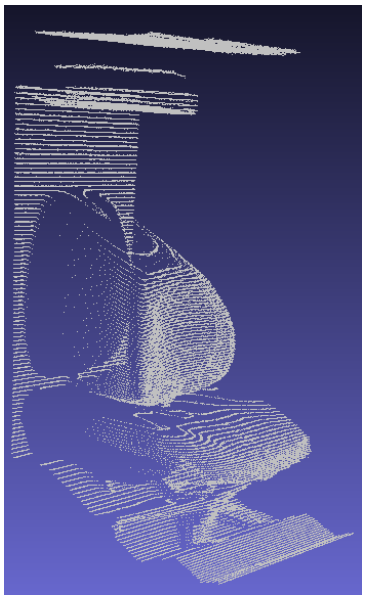
b) Raw 3D point cloud of the scene as captured by the Kinect (294237 points)



c) Raw 3D point cloud of the scene as captured by the ASLS (51592 points)



d) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (57440 points)



e) Raw 3D point cloud of the scene as captured by the SICK/Track (99052 points)

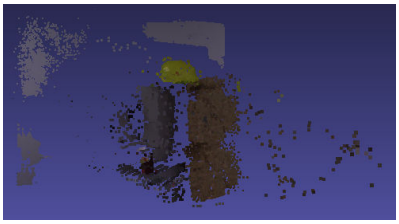
Figure C.9 - Scene 9: textured chair

C.10 Scene 10: Cluttered Stack

This scene, shown in Figure C.10, contains several objects stacked together. All the sensors were able to acquire this scene, at least in their respective fields of views. As with Section C.8, the track sensors performed two acquisitions such that each was able to acquire an almost complete scan. The objects all contain different surfaces, such as the wooden pallet, the plastic printer, the shiny yellow hard hat, the brown paper cup, and the metal bench, which all meet at different angles that provide interesting contours and other features for the analysis of the behaviour of the proposed techniques from Chapter 3.

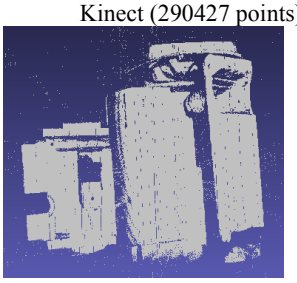
a) RGB image of the scene

b) Raw 3D point cloud of the scene as captured by the Kinect (290427 points)

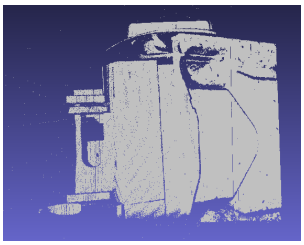


c) Raw 3D point cloud of the scene as captured by the ASLS (85028 points)

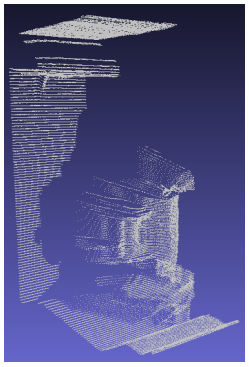
d) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (80006 points)



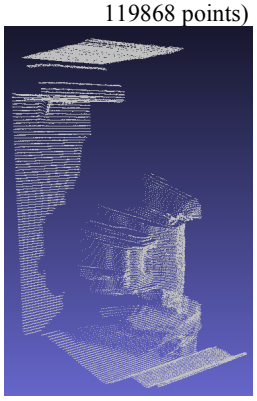
e) Raw 3D point cloud of the scene as captured by the Jupiter/Track (left side, 119868 points)



f) Raw 3D point cloud of the scene as captured by the Jupiter/Track (right side, 147674)



g) Raw 3D point cloud of the scene as captured by the SICK/Track (left side, 99314 points)



h) Raw 3D point cloud of the scene as captured by the SICK/Track (right side, 99311 points)

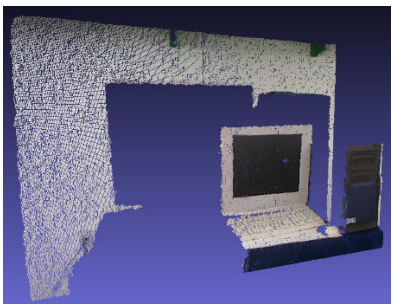
Figure C.10 - Scene 10: cluttered stack

C.11 Scene 11: Computer

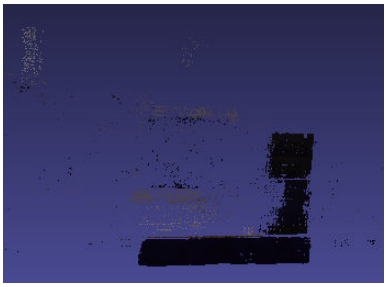
This scene, as shown in Figure C.11, of a computer system, was acquired by the Kinect, ASLS, Jupiter mounted on the F3T robotic arm, Jupiter mounted on the linear motorized track, and SICK LMS mounted on the linear motorized track. As previously performed in Sections C.8 and C.10, the Jupiter and SICK LMS mounted on the track captured two acquisitions with shifted viewpoints in order to fully capture the scene. The glass of the CRT monitor was not acquired fully by all the sensors, except for the SICK LMS. The keys from

the keyboard are an interesting small feature, which requires good scene coverage in order to detect, while the gaps of data along the monitor provide an interesting change in density for evaluating the proposed algorithms from Chapter 3.

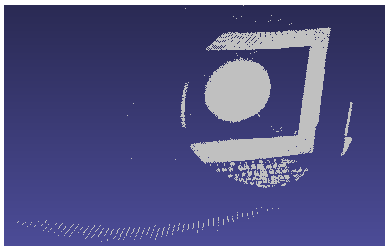
a) RGB image of the scene



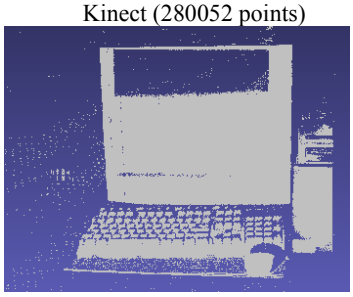
b) Raw 3D point cloud of the scene as captured by the Kinect (280052 points)



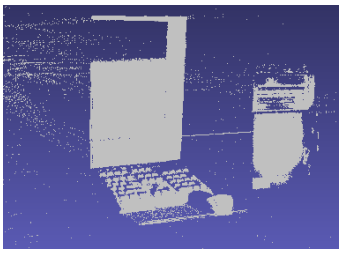
c) Raw 3D point cloud of the scene as captured by the ASLS (157776 points)



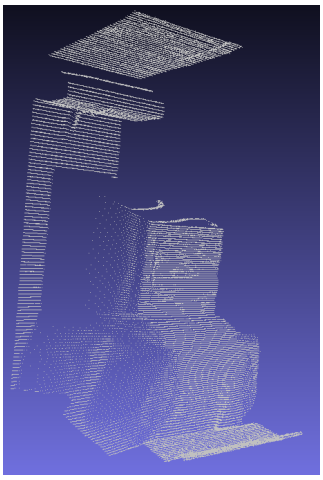
d) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (83805 points)



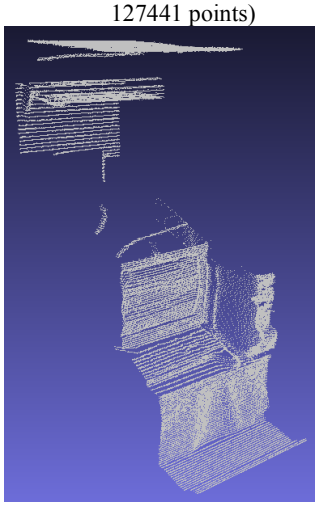
e) Raw 3D point cloud of the scene as captured by the Jupiter/Track (left side, 127441 points)



f) Raw 3D point cloud of the scene as captured by the Jupiter/Track (right side, 91012 points)



g) Raw 3D point cloud of the scene as captured by the SICK/Track (left side, 99550 points)



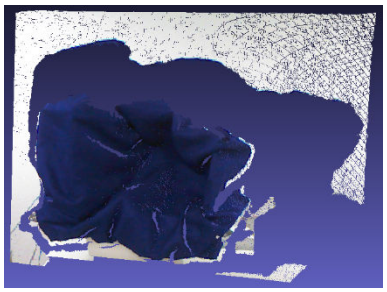
h) Raw 3D point cloud of the scene as captured by the SICK/Track (right side, 99504 points)

Figure C.11 - Scene 11: computer

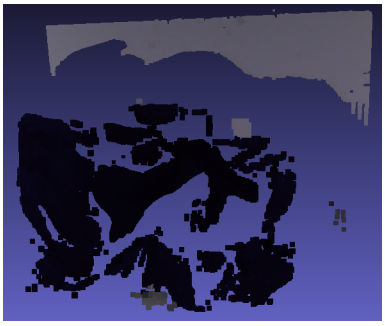
C.12 Scene 12: Fluidly Deformed Felt

This scene, shown in Figure C.12, consists of a few objects that are stacked and then covered with blue felt in order to create a fluidly deformed concave surface. The Kinect, ASLS, and the SICK LMS mounted on a linear motorized track were used to acquire the scene, as the red laser from the Jupiter sensor was unable to adequately reflect off of the felt surface. The fluidly curved surface contains many segments of convex and concave behaviour, which is of interest for the analysis of the algorithms presented in Chapter 3.

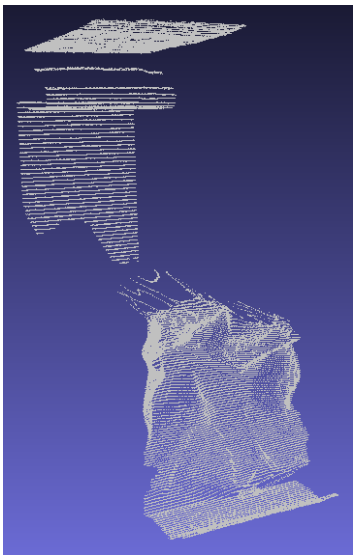
a) RGB image of the scene



b) Raw 3D point cloud of the scene as captured by the Kinect (285945 points)



c) Raw 3D point cloud of the scene as captured by the ASLS (208638 points)



d) Raw 3D point cloud of the scene as captured by the SICK/Track (99550 points)

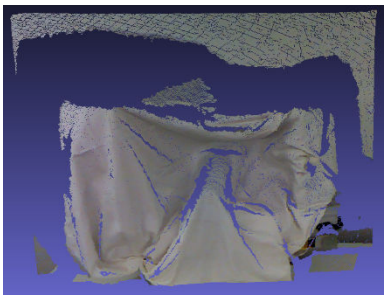
Figure C.12 - Scene 12: fluidly deformed felt

C.13 Scene 13: Fluidly Deformed Sheet

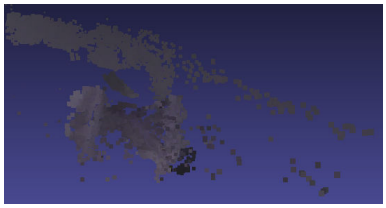
This scene, shown in Figure C.13, is similar to that presented in Figure C.12, but instead of blue felt being used to create the surface, a white sheet was used instead. The use of the white sheet allowed the Jupiter sensor to acquire points on the surface, and as a result, all the sensors were able to capture this scene. As was the case with scenes presented in Sections C.8, C.10, and C.11, the Jupiter and SICK LMS mounted on the linear motorized track captured the scene from two shifted perspectives in order to ensure that each sensor was able

to acquire the complete scene. The performance of the proposed techniques on this concave fluid scene is of interest for evaluating the techniques presented in Chapter 3.

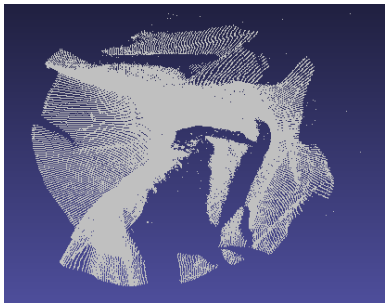
a) RGB image of the scene



b) Raw 3D point cloud of the scene as captured by the Kinect (284202 points)

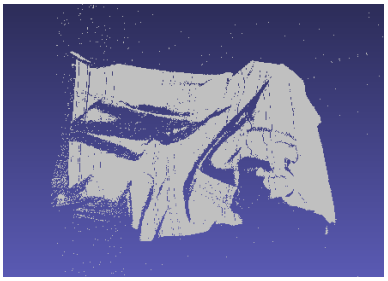


c) Raw 3D point cloud of the scene as captured by the ASLS (16953 points)

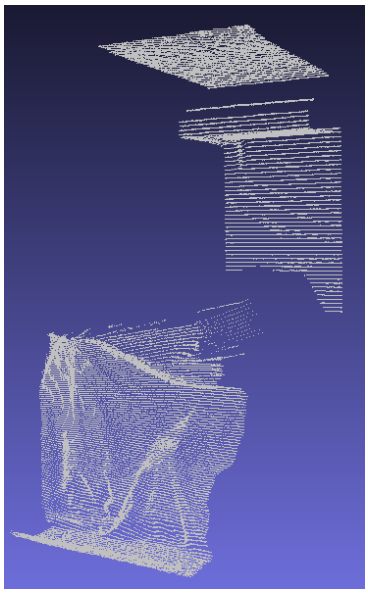


d) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (81471 points)

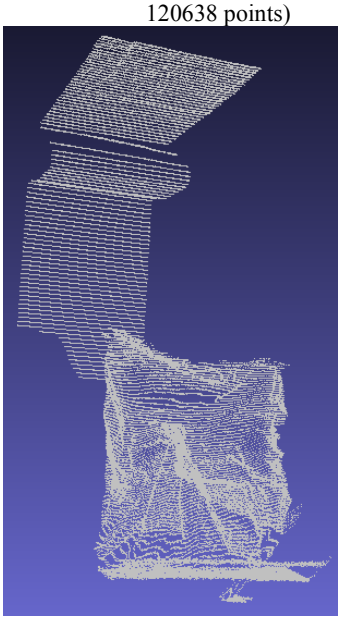
e) Raw 3D point cloud of the scene as captured by the Jupiter/Track (left side, 120638 points)



f) Raw 3D point cloud of the scene as captured by the Jupiter/Track (right side, 93579 points)



g) Raw 3D point cloud of the scene as captured by the SICK/Track (left side, 99550 points)



h) Raw 3D point cloud of the scene as captured by the SICK/Track (right side, 99550 points)

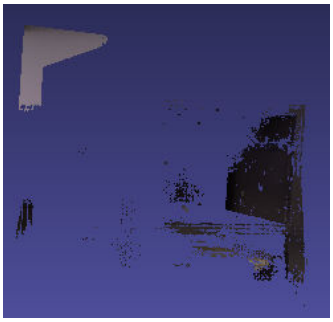
Figure C.13 - Scene 13: fluidly deformed sheet

C.14 Scene 14: Inspection Machine Angle 1

This scene, as presented in Figure C.14, contains an inspection machine, and was captured by the Kinect and the ASLS. The various perpendicular planes over a variety of depth produce an interesting scene. Both sensors, the ASLS more so than the Kinect, have difficulties with the metallic and black features of this scene.

a) RGB image of the scene

b) Raw 3D point cloud of the scene as captured by the Kinect (260103 points)

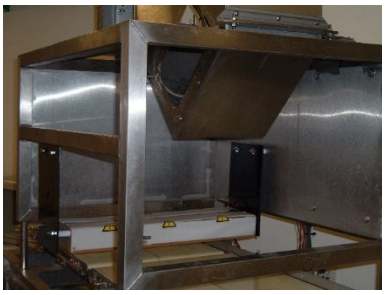


c) Raw 3D point cloud of the scene as captured by the ASLS (82966 points)

Figure C.14 - Scene 14: inspection machine angle 1

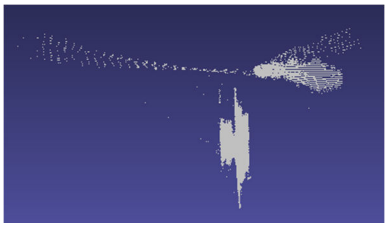
C.15 Scene 15: Inspection Machine Angle 2

This scene, as presented in Figure C.15, contains the same inspection machine as in Figure C.14, but presented at a different angle. The Kinect, Jupiter mounted on the F3T robotic arm, and SICK LMS mounted on the linear motorized track captured this scene. As with the scene in Section C.14, the various perpendicular planes over a variety of depth produce an interesting scene. The three sensors have difficulties with the metallic and black features of this scene.

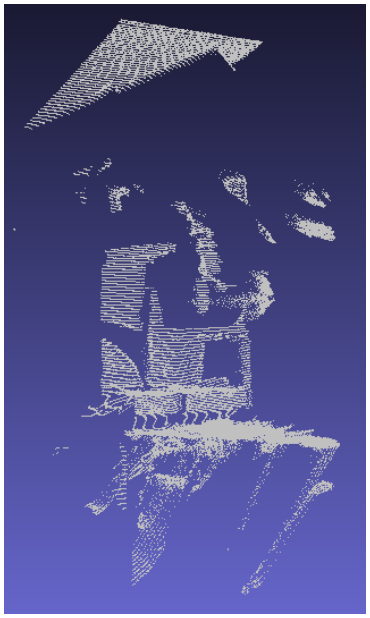


a) RGB image of the scene

b) Raw 3D point cloud of the scene as captured by the Kinect (263302 points)



c) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (30308 points)



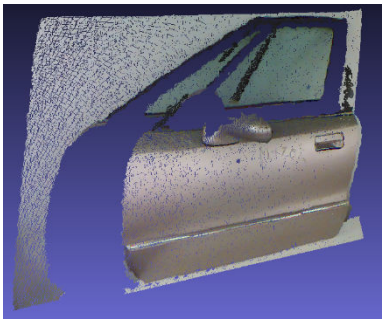
d) Raw 3D point cloud of the scene as captured by the SICK/Track (63658 points)

Figure C.15 - Scene 15: inspection machine angle 2

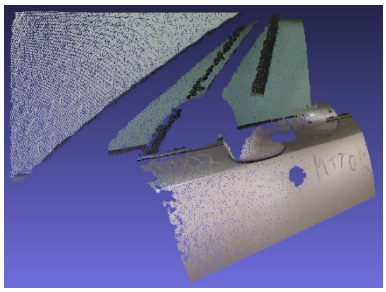
C.16 Scene 16: Car Door

This scene, shown in Figure C.16, contains a car door. The Kinect acquired the door from 3 different perspectives, one of the full door, one concentrating on the car door handle, and one concentrating on the car door mirror. In all three cases the window frame was not well captured, and there were some gaps along the main door panel. The ASLS acquired the full field of view of the car door, but the narrow segments of the window frame were not well captured, as well as the mid-lateral section of the door. Two acquisitions were performed using the Jupiter mounted on the F3T robotic arm, centered on the car door mirror and handle. The SICK LMS mounted on the linear motorized track acquired the car door, with the door frame and mirror poorly represented. The car door is interesting since all the sensors acquired the window frame poorly, which helps evaluate performance of the algorithms presented in Chapter 3. Furthermore the mirror provides a significant contusion from the otherwise near uniform curve of the surface of the door, and the handle provides several small scale features, such as the key hole, hand grip, and outlining indented contour.

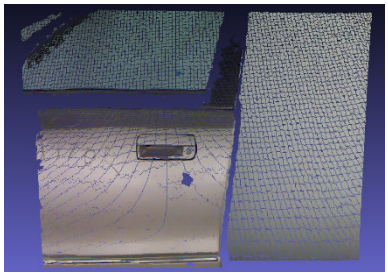
a) RGB image of the scene



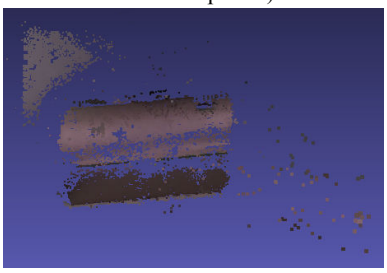
b) Raw 3D point cloud of the scene as captured by the Kinect (full fov, 283598 points)



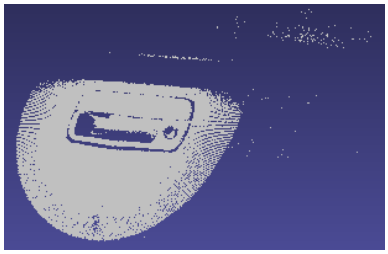
c) Raw 3D point cloud of the scene as captured by the Kinect (mirror, 273322 points)



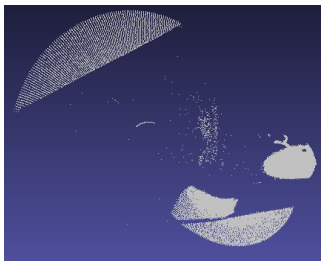
d) Raw 3D point cloud of the scene as captured by the Kinect (handle, 293034 points)



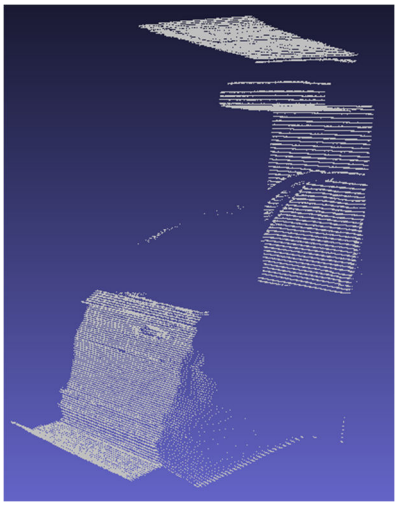
e) Raw 3D point cloud of the scene as captured by the ASLS (92044 points)



f) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (mirror, 41174 points)



g) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (handle, 44149 points)



h) Raw 3D point cloud of the scene as captured by the SICK/Track (97222 points)

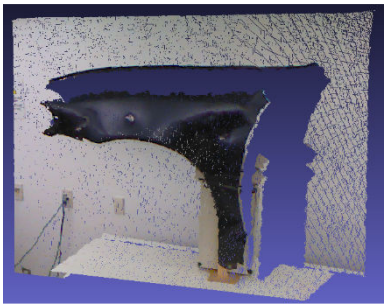
Figure C.16 - Scene 16: car door

C.17 Scene 17: Car Fender

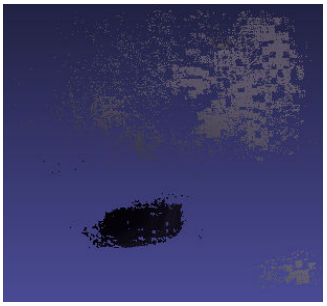
This scene, presented in Figure C.17, contains a black car fender with several dents present (the metallic silver area within the black fender). This scene was captured using the Kinect, ASLS, Jupiter mounted on the F3T robotic arm, and the SICK LMS mounted on the linear motorized track. The latter two sensors were used to acquire the scene from two shifted perspectives, due to the large scale of the scene. The Jupiter and ASLS perform poorly on this scene, with only a few parts of the fender being acquired. The SICK LMS acquired the fender well, although the segment by the lower wheel well has some data missing. The

fender provides an interesting scene due to the dents, and various curves present on the object.

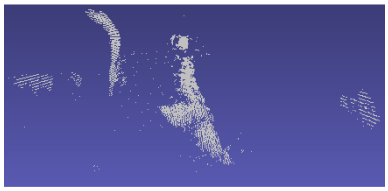
a) RGB image of the scene



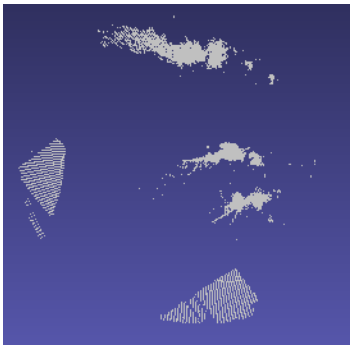
b) Raw 3D point cloud of the scene as captured by the Kinect (294664 points)



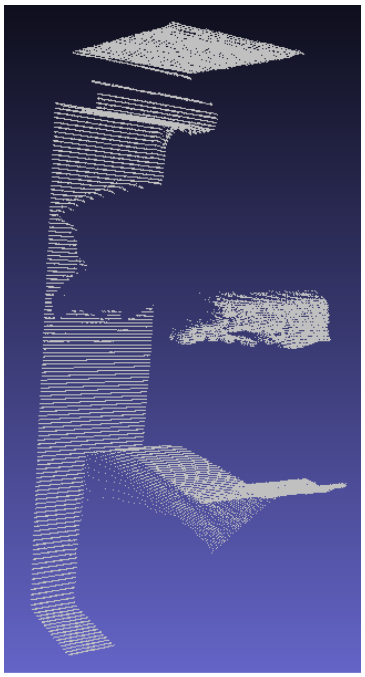
c) Raw 3D point cloud of the scene as captured by the ASLS (33204 points)



d) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (front, 70552 points)



e) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (rear, 90280 points)



f) Raw 3D point cloud of the scene as captured by the SICK/Track (front, 95664 points)

g) Raw 3D point cloud of the scene as captured by the SICK/Track (rear, 91897 points)

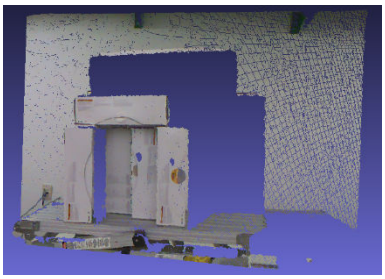
Figure C.17 - Scene 17: car fender

C.18 Scene 18: Stacked Boxes

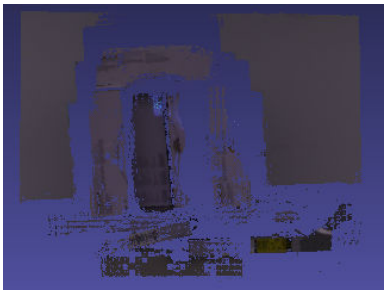
This scene, as seen in Figure C.18, contains four Kinect boxes arranged vertically at different depths, with a fifth stacked horizontally across the top. The Kinect, ASLS, Jupiter mounted on the F3T robotic arm, Jupiter mounted on the linear motorized track, and SICK LMS mounted on the linear motorized track captured the scene. The different depths of the boxes

within an enclosed environment create scene that is useful for evaluating performance of the algorithms presented in Chapter 3 in regards to different discontinuity distances.

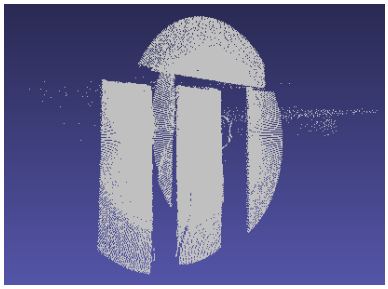
a) RGB image of the scene



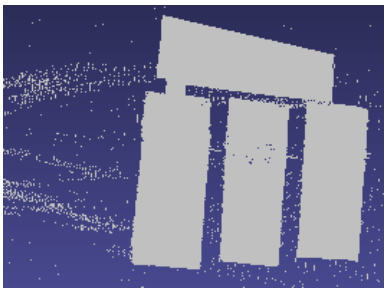
b) Raw 3D point cloud of the scene as captured by the Kinect (283719 points)



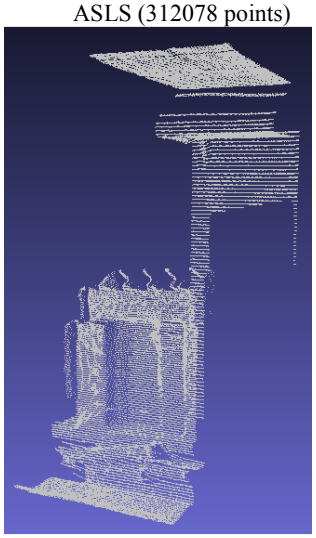
c) Raw 3D point cloud of the scene as captured by the ASLS (312078 points)



d) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (91996 points)



e) Raw 3D point cloud of the scene as captured by the Jupiter/Track (151194 points)

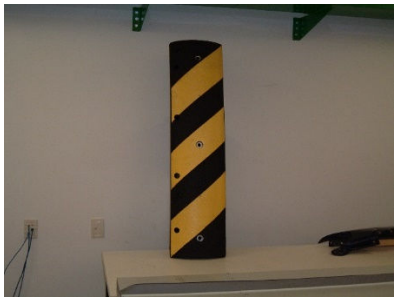


f) Raw 3D point cloud of the scene as captured by the SICK/Track (99160 points)

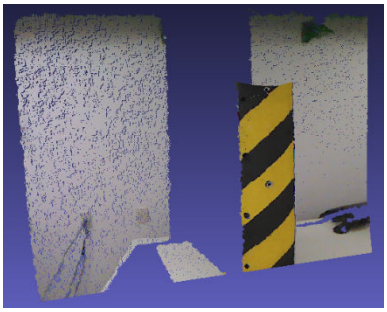
Figure C.18 - Scene 18: stacked boxes

C.19 Scene 19: Rubber Speed Bump

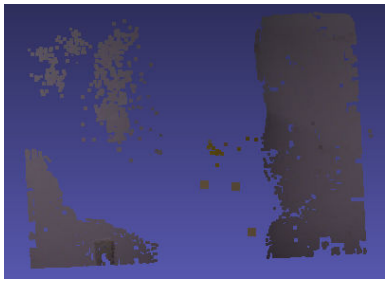
This scene, shown in Figure C.19, contains a rubber black and yellow speed bump that is stood vertically on its end. The Kinect, ASLS, Jupiter mounted on the F3T robotic arm, and SICK LMS mounted on the linear motorized track acquired this scene. All of the sensors, except for the Kinect, had difficulties acquiring the black segments of the speed bump, with the ASLS having the most difficulty due to the reflective nature of the yellow paint, and absorptive nature of the black rubber in providing an inadequate surface on which the projected pattern can be detected.



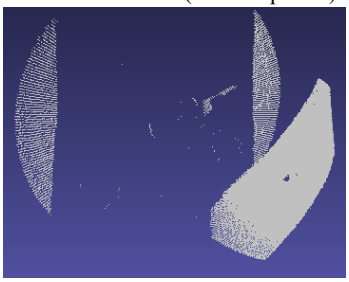
a) RGB image of the scene



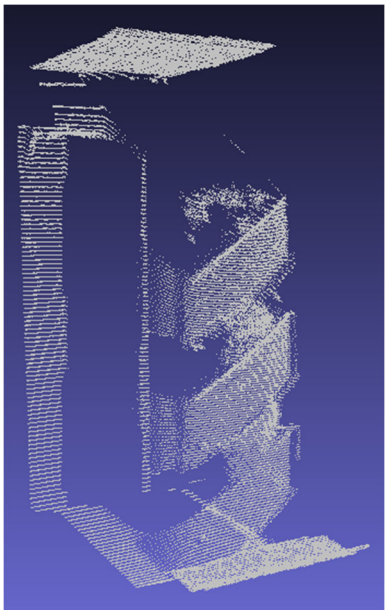
b) Raw 3D point cloud of the scene as captured by the Kinect (291579 points)



c) Raw 3D point cloud of the scene as captured by the ASLS (123677 points)



d) Raw 3D point cloud of the scene as captured by the Jupiter/F3T (49591 points)

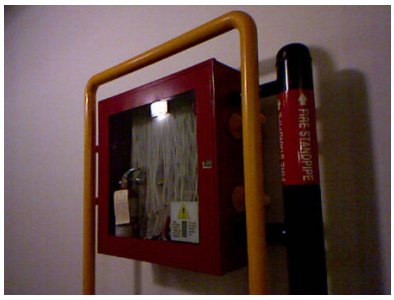


e) Raw 3D point cloud of the scene as captured by the SICK/Track (88830 points)

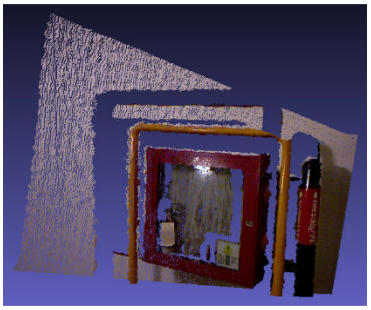
Figure C.19 - Scene 19: rubber speed bump

C.20 Scene 20: Fire Hose Station

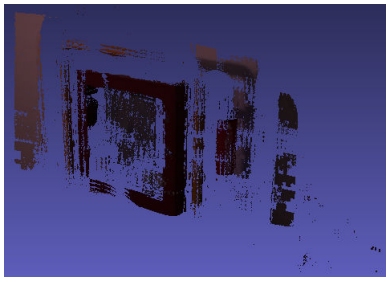
This scene, shown in Figure C.20, contains a fire hose station that was captured using the Kinect, ASLS, Jupiter mounted on the linear motorized track, and SICK mounted on the linear motorized track. The fire hose station contains interesting folds of the hose behind the glass, in addition to the surrounding metal frame. The Jupiter and ASLS have some issues due to reflections and refractions of their laser/projected patterns on the glass.



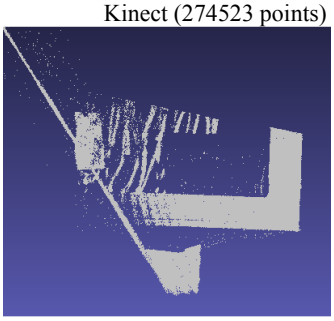
a) RGB image of the scene



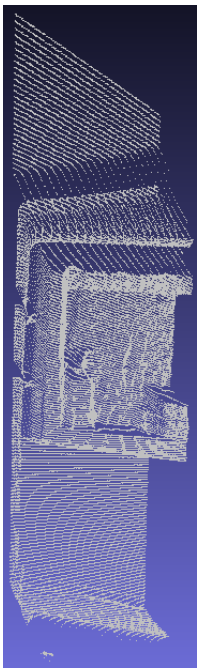
b) Raw 3D point cloud of the scene as captured by the Kinect (274523 points)



c) Raw 3D point cloud of the scene as captured by the ASLS (205778 points)



d) Raw 3D point cloud of the scene as captured by the Jupiter/Track (82716 points)

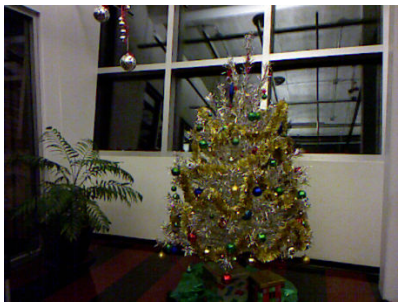


e) Raw 3D point cloud of the scene as captured by the SICK/Track (93989 points)

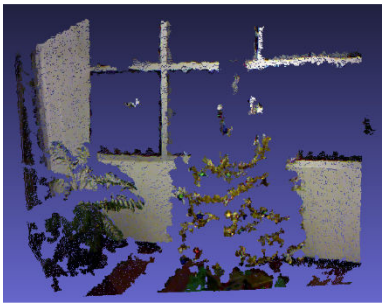
Figure C.20 - Scene 20: fire hose station

C.21 Scene 21: Christmas Tree

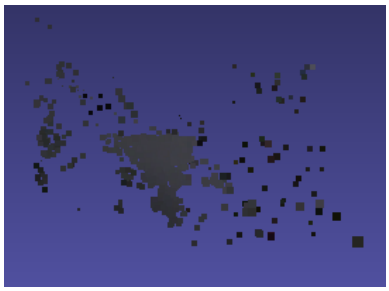
This scene, presented in Figure C.21, contains a decorated metallic Christmas tree that was captured using the Kinect, ASLS, and the SICK LMS mounted on the linear motorized track. The ASLS performed poorly, as there was not many sections on the tree that were large enough on which a single 3x3 pattern could be projected. The Kinect was only able to acquire some of the balls and the gold garland decorating the tree. The SICK performed a bit better, capturing the larger scale features of the tree. The many small projections of the tree provide an interesting surface to capture, and the resulting 3D points clouds that are sparse also provide interesting datasets in order to analyse the performance of the algorithms presented in Chapter 3 in regards to performance on largely unacquirable surfaces.



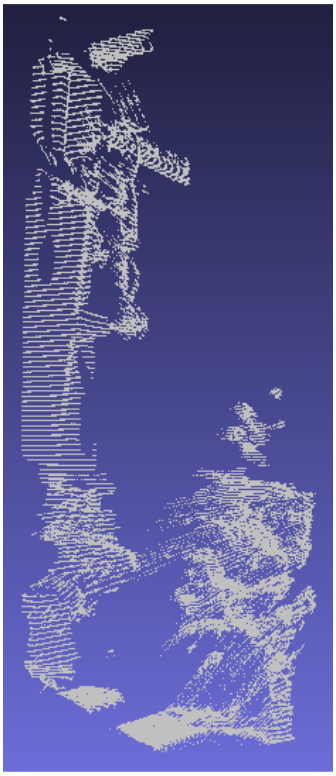
a) RGB image of the scene



b) Raw 3D point cloud of the scene as captured by the Kinect (150275 points)



c) Raw 3D point cloud of the scene as captured by the ASLS (1503 points)



d) Raw 3D point cloud of the scene as captured by the SICK/Track (91935 points)

Figure C.21 - Scene 21: Christmas tree

C.22 Scene 22: Car Door Segment

This scene of the car door segment, as shown in Figure C.22, is acquired by the Neptec LMS. This door segment contains the door seam, in addition to the handle. The high precision dense acquisition provided by the Neptec LMS contributes another rich scene captured by an additional sensor on which the performance of the developed techniques from Chapter 3 can be evaluated.

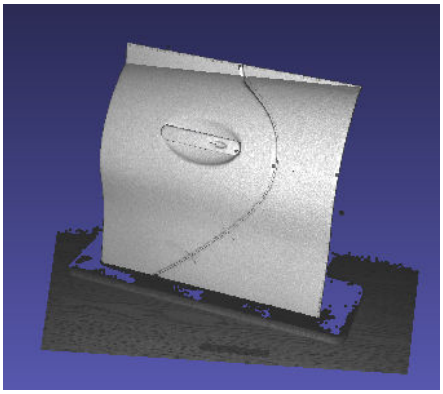


Figure C.22 - Scene 22: car door segment – raw 3D point cloud as captured by the Neptec LMS (695122 points)

Appendix D. Additional Results

This appendix provides additional results and analysis from Chapter 5, many of which confirm, complement, and expand upon the evaluation.

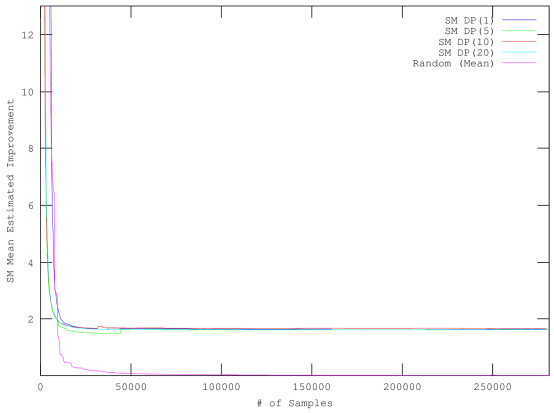
D.1 Additional Quantitative Analysis of the Surface Mesh Improvement Based Selection Process

This section provides additional analysis for Scenes 2, 4 and 5 that expand upon that presented in Section 5.2.1.

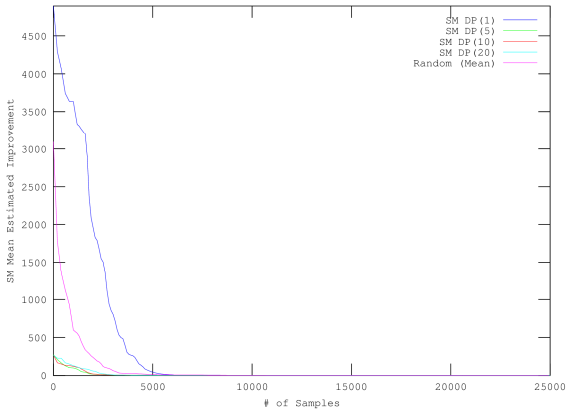
D.1.1 Analysis of Scene 2

This section provides the analysis of the techniques as performed on Scene 2 (see Section C.2). Figure D.1 a) and b) shows that the surface mesh mean estimated improvement when using the surface mesh improvement based selection process converges more rapidly than random sampling, except for the case of SM DP(1), but random sampling converges to a lower improvement level. As will be seen in Figure D.6 which is presented in Section D.5.1, the fact that SM DP(1) converges to the minimum slower than random point selection and the multipoint techniques is due to a zigzag pattern being executed by the selection process at an early stage. This behaviour will be examined more closely in Section D.5.1, where it is put into evidence. The selection process for the other values of N, that is when 5, 10, and 20 points are selected at the same time, avoids the negative effect of this behavior. As for the random point selection process converging to a slightly lower improvement level, this is likely due to minor differences in the Delaunay triangulation created when the randomly distributed samples are merged into the mesh, instead of points selected by the proposed surface mesh improvement based approach. It should be additionally noted that convergence to the minimum occurs slightly after 5000 points have been inserted for all the curves, as put into evidence in Figure D.1 b). The surface mesh estimated improvement errors in Figure D.1 c) and d) show that the surface mesh improvement based selection process when choosing multiple points at each iteration, converges to the minimum earlier than the random sampling and SM DP(1). This behaviour of the single point surface mesh improvement based selection process of slow convergence to the minimum value is as previously mentioned due to the nature of the zigzagged initial acquisition. It should be noted that even though it converges much slower, it still occurs prior to 5000 points being inserted into the

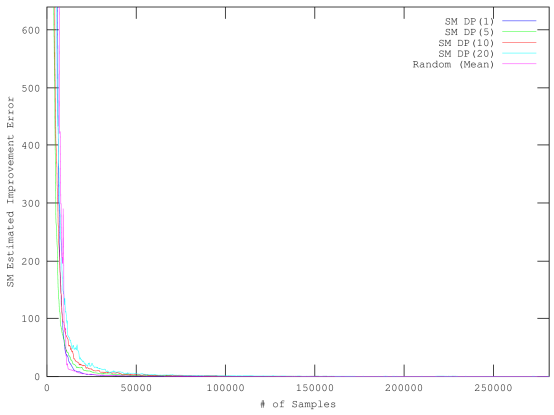
model. Figure D.1 e) exhibits the same behaviour as that in Figure 5.3 e), again verifying that the surface mesh improvement based selection process favours accuracy instead of coverage.



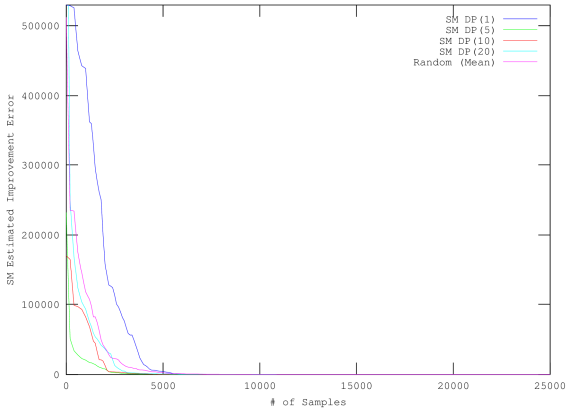
a) Surface mesh mean estimated improvement over all points



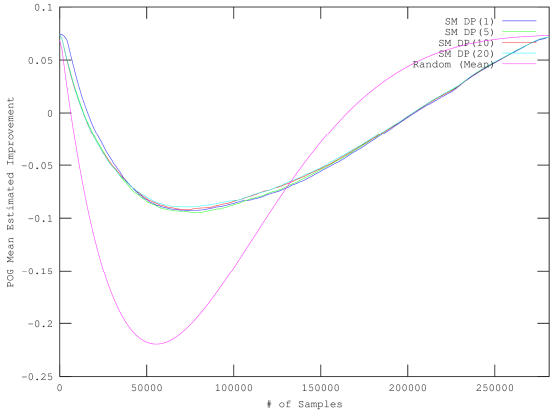
b) Surface mesh mean estimated improvement over 25000 points



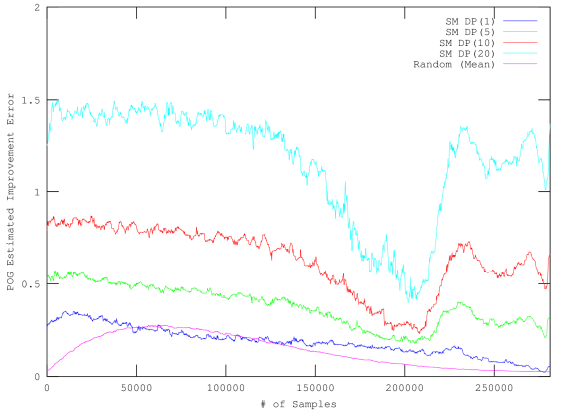
c) Surface mesh estimated improvement error over all points



d) Surface mesh estimated improvement error over 25000 points



e) Probabilistic occupancy grid mean estimated improvement over all points

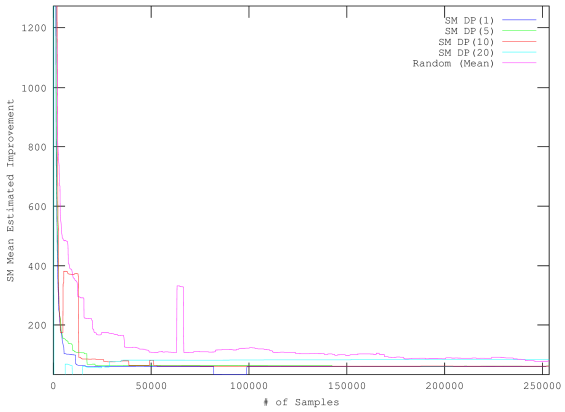


f) Probabilistic occupancy grid estimated improvement error over all points

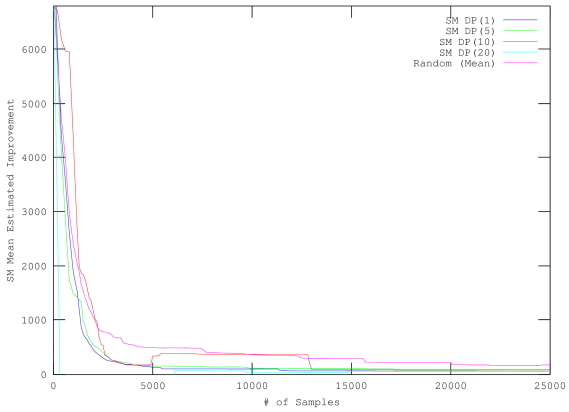
Figure D.1 - Scene 2: mean improvement and improvement error trend graphs

D.1.2 Analysis of Scene 4

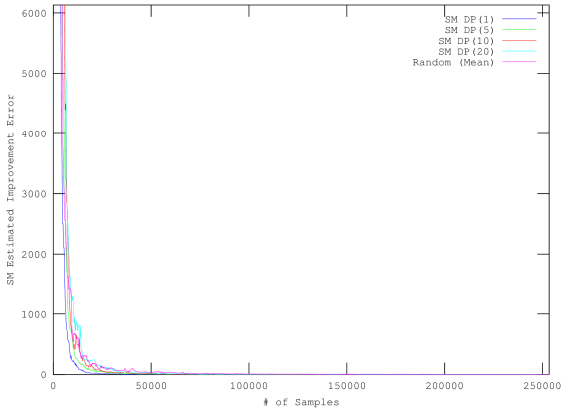
This section details the analysis of the techniques as they operated on Scene 4 (see Section C.4). Figure D.2 a) and b) which shows the surface mesh mean estimated improvement, indicates that the surface mesh improvement based selection processes converge to their respective minimum values with much fewer points inserted into the model than does the case where random point selection drives the point acquisition. Convergence to the minimum for this scene, which is put into evidence by Figure D.2 b), occurs after the insertion of 5000-10000 points, except for the cases of random point selection, and SM DP (10). The curve representing SM DP(10) exhibits a small increase that covers the section between 5000 and 13000 points inserted, while the random point selection shows a gradual decay to the minimum value after 5000 points have been inserted into the model. The surface mesh estimated improvement error, as shown in Figure D.2 c) and d), demonstrates that the respective curves for this error converge to the minimum value prior to 5000 points being inserted, except for the case of SM DP(20) that converges to the minimum value slightly after 5000 points have been inserted into the model. As put into evidence by Figure D.2 d), the surface mesh improvement based selection process where N=1 (SM DP(1)) converges with fewer points being inserted into the model than the other methods. Figure D.2 e) shows similar behaviour for the probabilistic occupancy grid mean estimated improvement as previously shown in all the earlier scenes, namely that the random point selection process has a much lower minimum than the surface mesh improvement based selection processes, which indicates that it provides better coverage, with this minimum occurring after approximately 50000 points have been inserted. Furthermore, all the surface mesh improvement based selection processes exhibit near identical curves. Finally, Figure D.2 f), which details the progression of the probabilistic occupancy grid estimated improvement error metric as points are being inserted into the model, shows that the random point selection process has a lower value for this error most of the time than that produced by the surface mesh improvement based selection process, indicating as previously observed that random point selection provides better scene coverage.



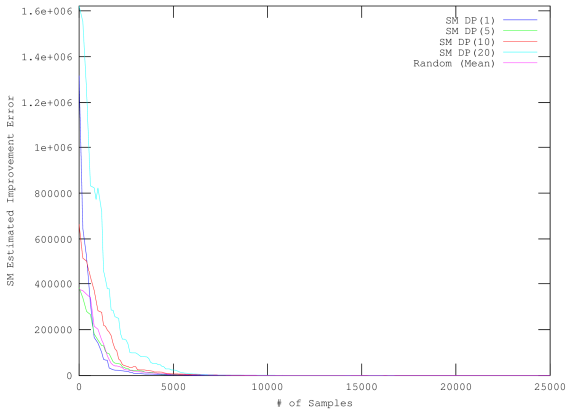
a) Surface mesh mean estimated improvement over all points



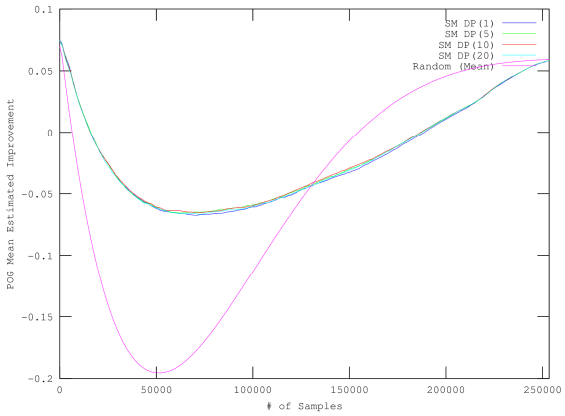
b) Surface mesh mean estimated improvement over 25000 points



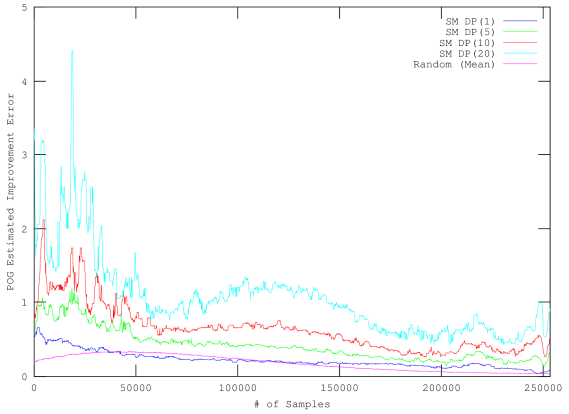
c) Surface mesh estimated improvement error over all points



d) Surface mesh estimated improvement error over 25000 points



e) Probabilistic occupancy grid mean estimated improvement over all points

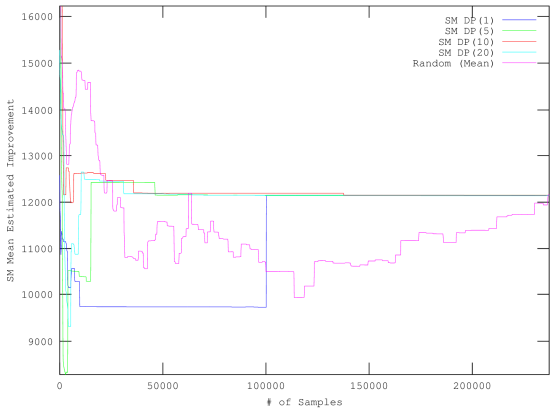


f) Probabilistic occupancy grid estimated improvement error over all points

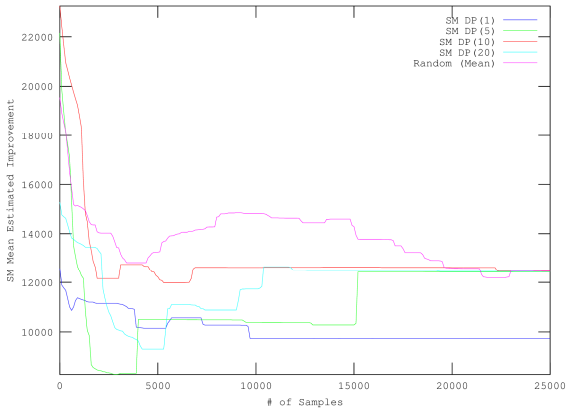
Figure D.2 - Scene 4: mean improvement and improvement error trend graphs

D.1.3 Analysis of Scene 5

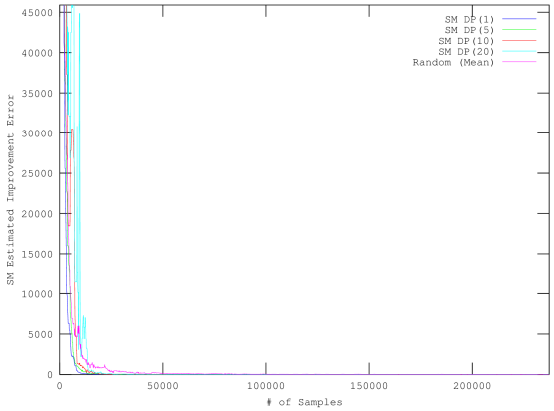
This section provides the analysis of the techniques as performed on Scene 5, shown in Section C.5. Figure D.3 a) and b) show that the surface mesh mean estimated improvement when using the surface mesh improvement based selection process, for all N tested, converges to the minimum level with fewer points inserted than when guided by random point selection. For this particular scene, the results appear to be noisy, with many ups and downs prior to the convergence at a mean estimated improvement level of approximately 12300. This is due to the large amounts of empty space in the scene (beyond the reach of the Kinect sensor employed for this case), as well as the pillars and pay machine being small, and at different depths, on which the initial sparse sampling does not pick up a substantial number of points. But despite these drawbacks, convergence still occurs prior the 15000 point mark for the surface mesh improvement based selection process when (N=5,10,20), but does not occur until the 100000 point mark for N=1, and much later for the case of random point selection. This is likely due to the surface mesh improvement based selection process for when N=1 scanning in more local regions than when multiple points are used, and hence it takes more points to determine the extent of the objects in the scene. Figure D.3 c) and d) shows that convergence to the minimum surface mesh estimated improvement error occurs after fewer points have been inserted into the model when using the surface mesh improvement based selection process (for N=1) than does random sampling, although there is much noise present in the plots, again due to the nature of the scene. Furthermore, as put into evidence in Figure D.3 d), the convergence to this minimum value occurs when 500-1000 points have been inserted into the model. Figure D.3 e) exhibits the similar behaviour for the probabilistic occupancy grid mean estimated improvement as that shown in the previous scenes, again verifying that the surface mesh improvement based selection process favours accuracy instead of coverage, while random point selection favours coverage, with the minimum of the curve occurring after approximately 50000 have been inserted into the model. Similarly, the behaviour of the selection processes on the probabilistic occupancy grid estimated improvement error metric, shown in Figure D.3 f), indicates that the random point selection process has better coverage than does the surface mesh improvement based selection processes.



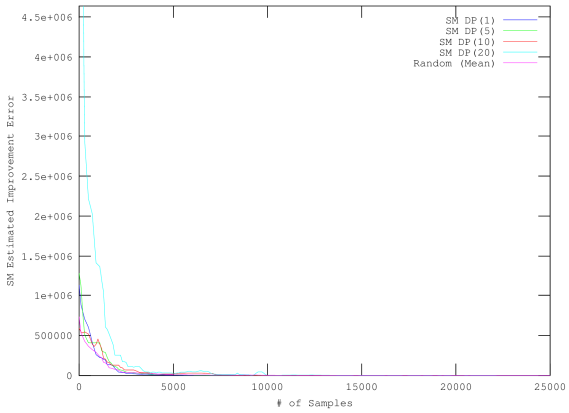
a) Surface mesh mean estimated improvement over all points



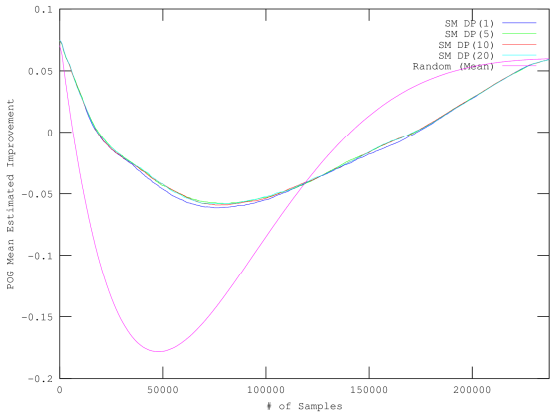
b) Surface mesh mean estimated improvement over 25000 points



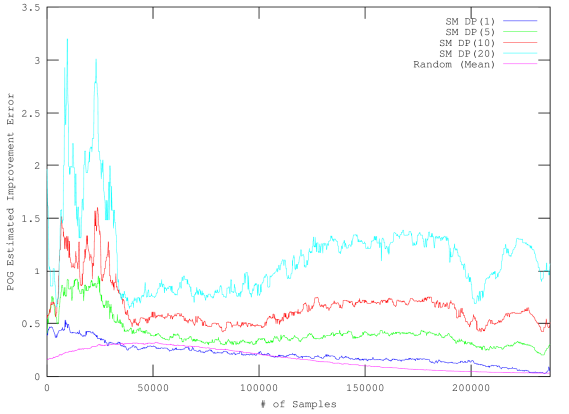
c) Surface mesh estimated improvement error over all points



d) Surface mesh estimated improvement error over 25000 points



e) Probabilistic occupancy grid mean estimated improvement over all points



f) Probabilistic occupancy grid estimated improvement error over all points

Figure D.3 - Scene 5: mean improvement and improvement error trend graphs

D.2 Additional Progression Results and Analysis of the Surface Mesh Improvement Based Selection Process

The results demonstrating and analysing the progression for Scenes 2, 4, and 5 are presented in this section, as they are very much similar to the three other scenes presented in Section 5.2.2.

D.2.1 Scene 2 - Empty Parking Garage Corner

In Scene 2 (shown in Section C.2), the empty parking garage corner, the surface mesh improvement based selection process appears to first focus on the bottom left part of the scene, where the floor meets the wall. This behaviour, put into evidence in Figure D.6, appears to be due to there being no valid points acquired be the Kinect sensor present in the initial 10x10 sparse subsampling down the extreme left of the image, as evidenced by the large partial white triangle in Figure D.4 a) indicating maximum potential improvement in this region. Furthermore, from the progression shown in Figure D.6, it appears that the surface mesh improvement based selection process chooses the majority of the points to acquire early on to follow a vertical zigzag pattern as a result of this maximum improvement. This behaviour explains the results from the graphs in Figure D.1 a) and b) where the random point selection process initially progressed more rapidly towards the convergence point for the surface mesh mean estimated improvement metric than the single point surface mesh improvement based selection process. Among the points acquired outside this region of zigzag acquisition, the majority tend to be in regions where the wall meets the ground (shown by the whiter triangles in the bottom third of the scene in Figure D.4), as well as on the ground (which is lumpy and therefore featured, while the walls are flat). Overall, the surface mesh improvement based selection process chooses the areas with the highest estimated improvement, which tend to correspond to larger areas, as well as areas where there is a possible transition. A fairly accurate representation of depth is therefore obtained after about 5000 points are acquired, as shown in Figure D.5 e), which matches well with the quantitative analysis for the convergence of the surface mesh mean estimated improvement shown in Figure D.1 b), and for the convergence of the surface mesh estimated improvement error shown in Figure D.1 d).

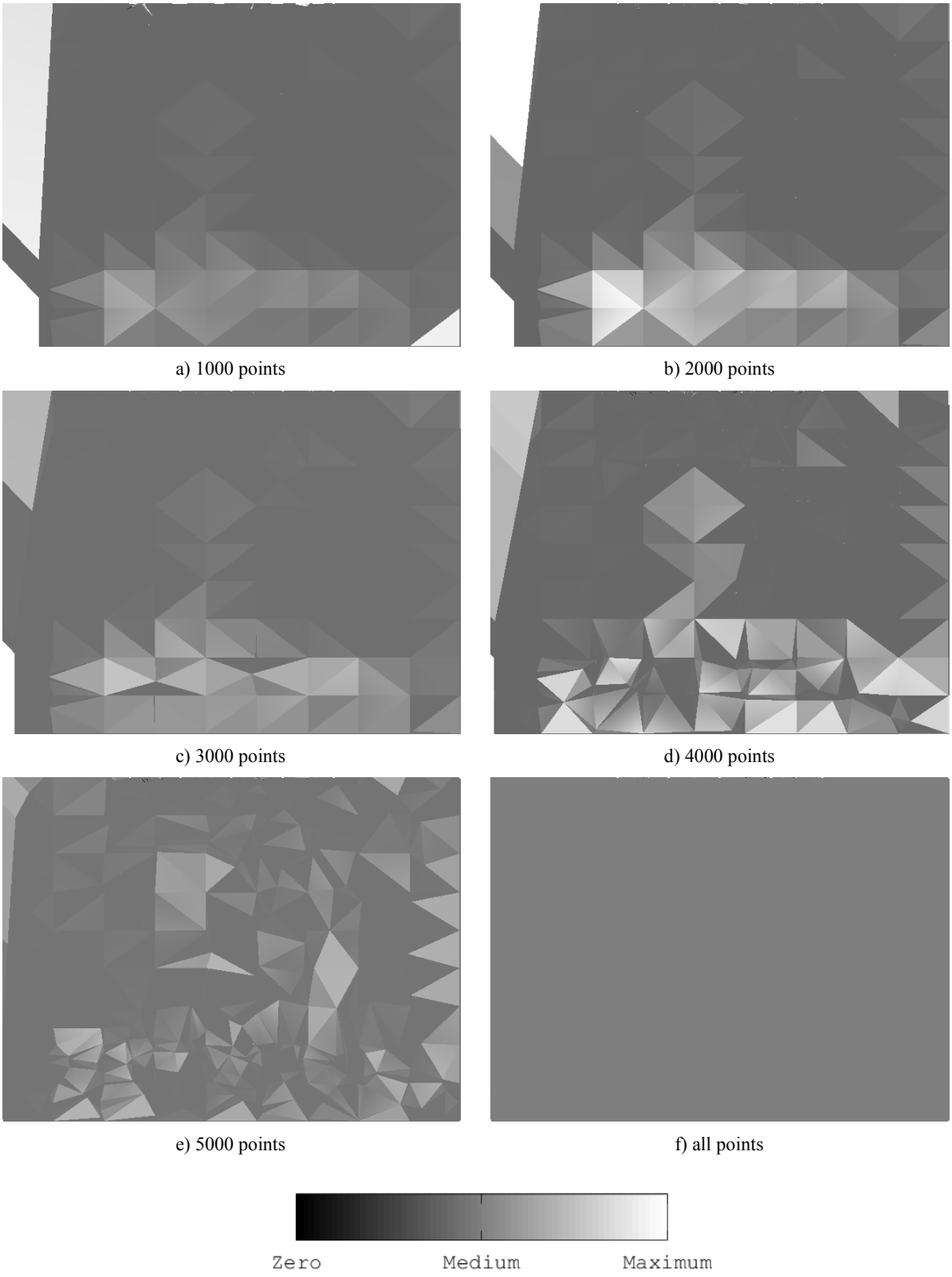


Figure D.4 - Scene 2: surface mesh improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

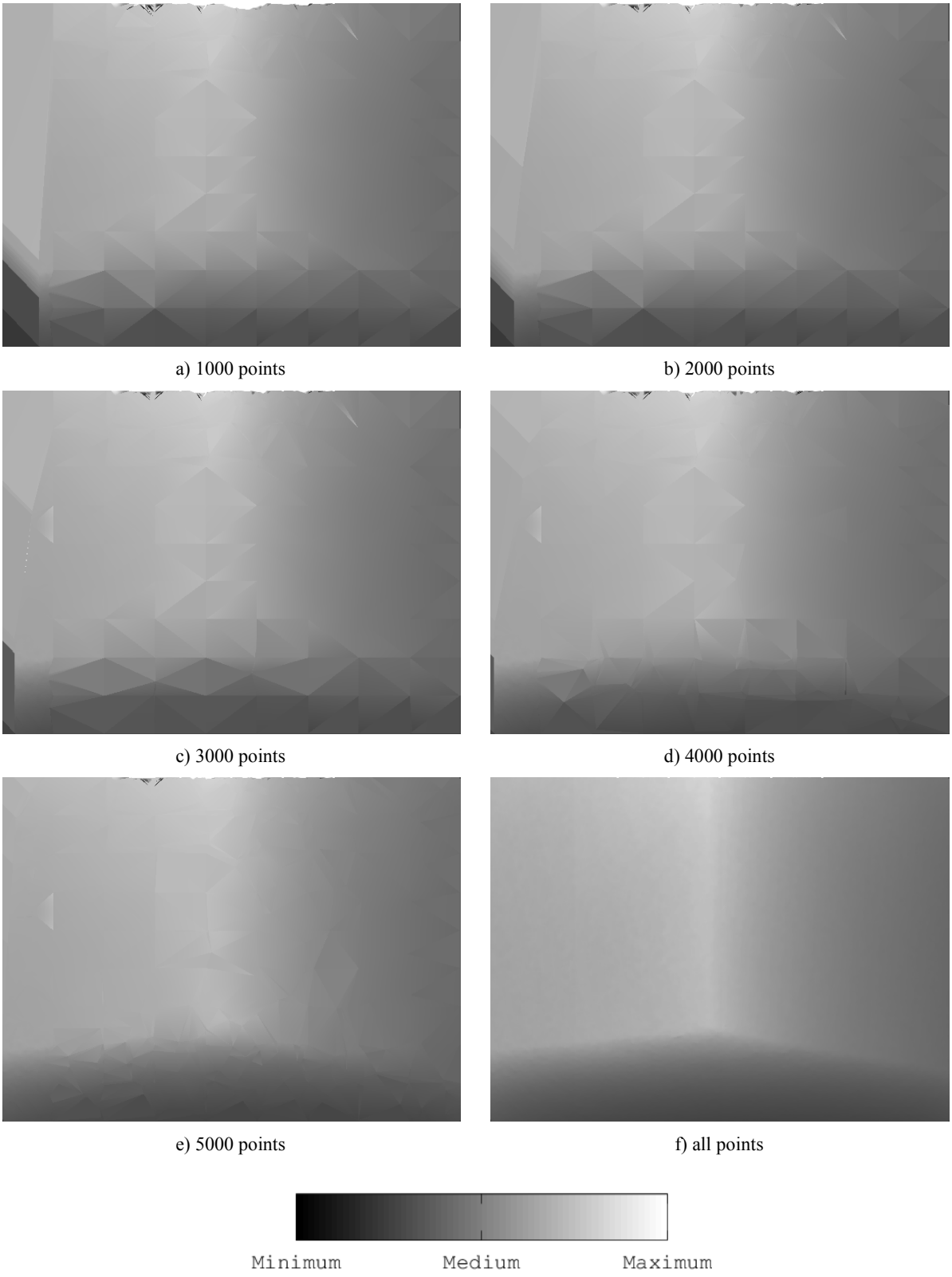


Figure D.5 - Scene 2: surface mesh based depth map progression after different numbers of points are acquired using the surface mesh improvement based selection process

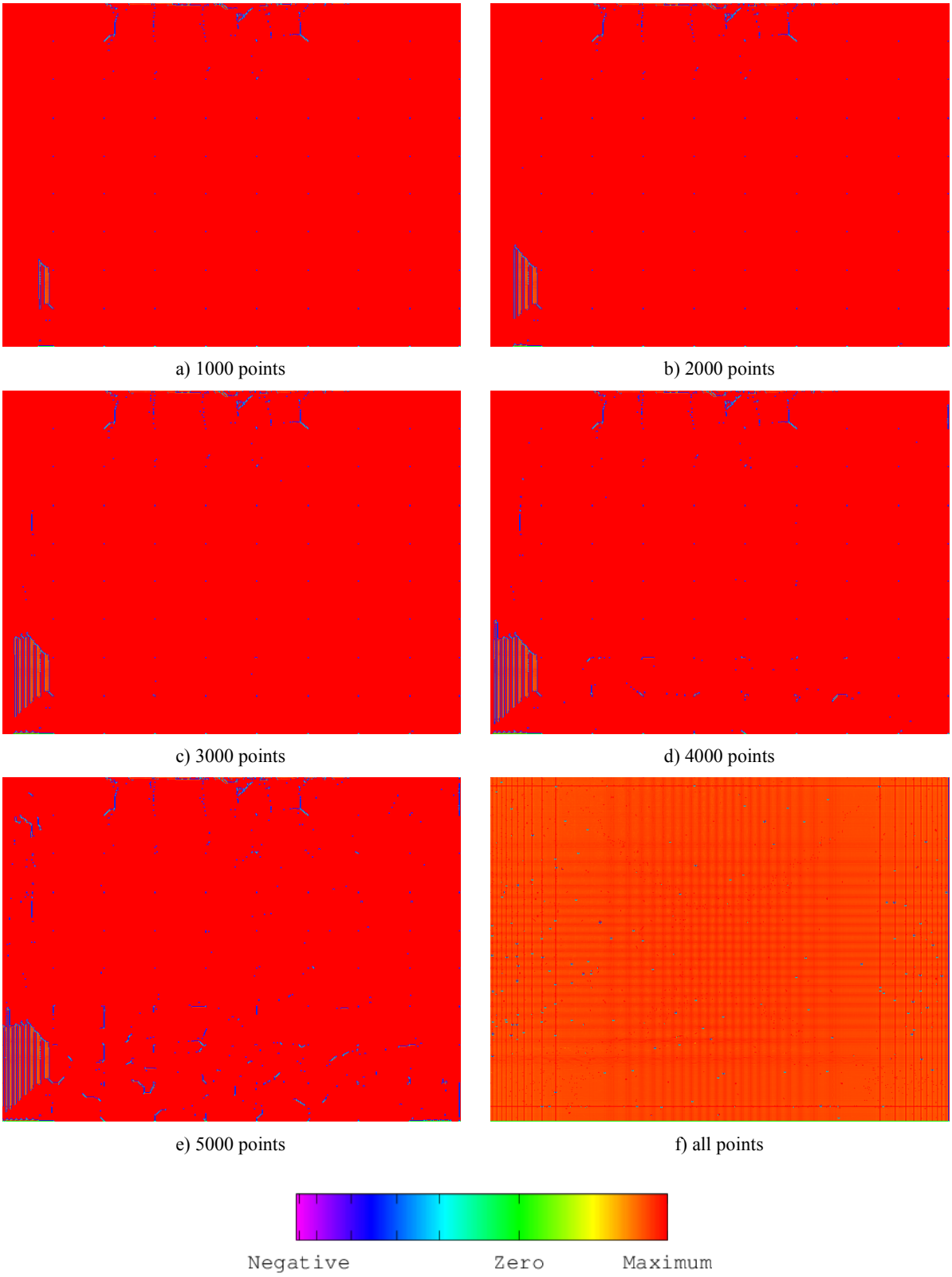


Figure D.6 - Scene 2: probabilistic occupancy grid improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

D.2.2 Scene 4 - Car and Wall in a Parking Garage

The progression of Scene 4, consisting of the car beside a wall in a parking garage (Section C.4), is shown in Figure D.7. In the first case, Figure D.7 a), represents the state of the surface mesh improvement progression after 1000 points have been selectively inserted by the proposed surface mesh improvement based selection processes. It illustrates that the edge of the car as well as the region between the parallel and back walls have been assigned a high improvement value (white/light gray triangles), while the more planar regions that exhibit little depth variance have lower improvement values (darker triangles). Progressive samples, directed by the surface mesh improvement based selection process, are concentrated in these transition regions (Figure D.7 b)) creating many small triangles, until the potential improvement of points acquirable by the Kinect sensor is low enough, then the selection process starts investigating the larger scale features along the wall (Figure D.7 b),c),d)). Figure D.7 b) is especially good at showing that the surface mesh improvement based selection process is selecting points within triangles covering a transition (smaller white triangles around the outside edge of the car), as well as points within larger triangles covering flatter regions with a relatively low transition value, with the result ending up being smaller triangles for the transition regions and larger triangles for flatter regions having roughly equivalent improvement values (identified by equivalent gray levels). The observations of the acquisition pattern for this scene, as is put into evidence in Figure D.9 by the blue regions of negative improvement that are present around the edges of acquired points, match clearly with those inferred from the surface mesh improvement map (Figure D.7). Furthermore most of this scene is recognizable by only 3000 points, as is observed in Figure D.8 c), which matches the quantitative result of where the convergence of the surface mesh mean estimated improvement occurs in Figure D.2 b), as well as where the convergence of the surface mesh estimated improvement error occurs in Figure D.2 d).

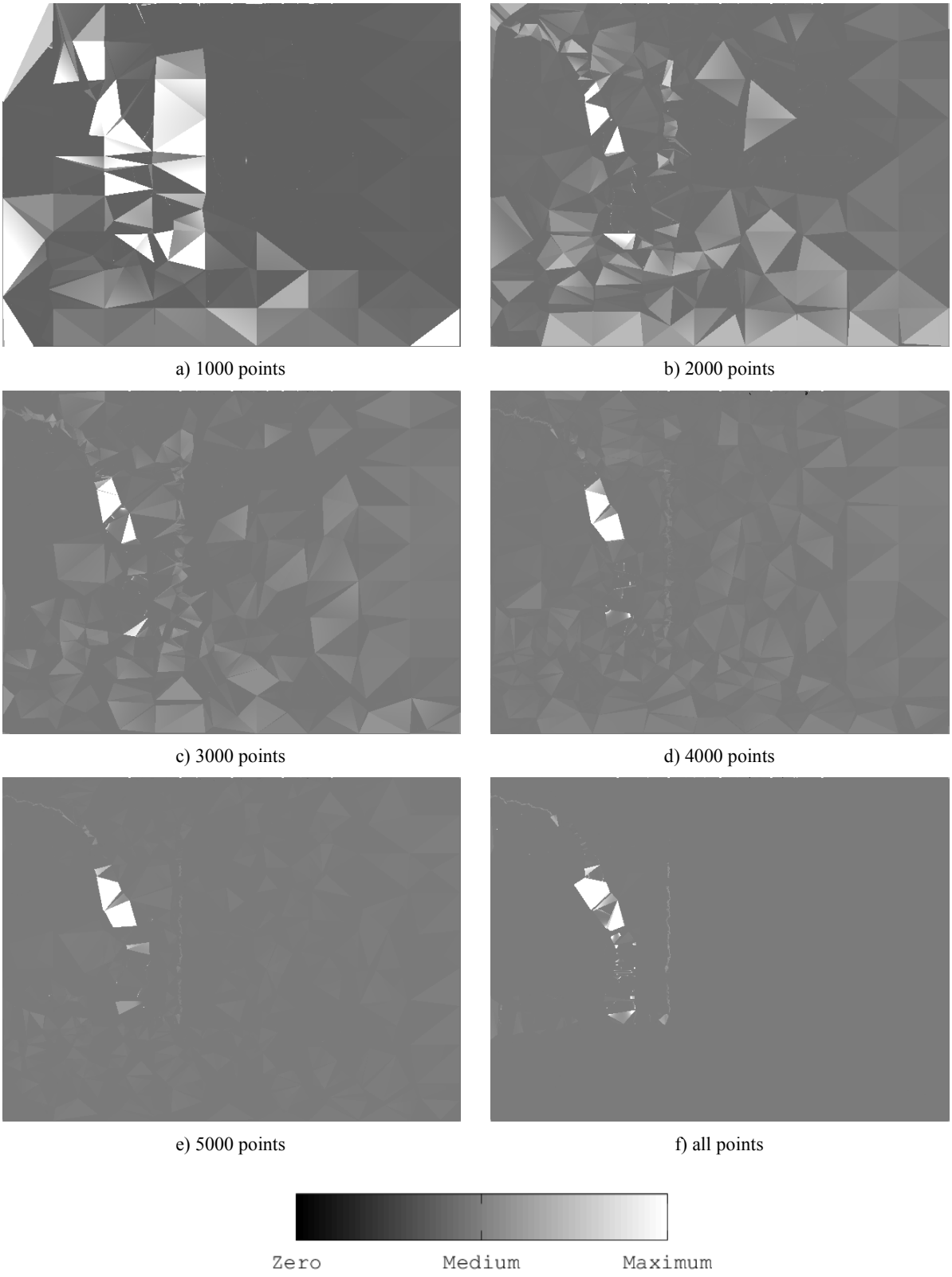


Figure D.7 - Scene 4: surface mesh improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

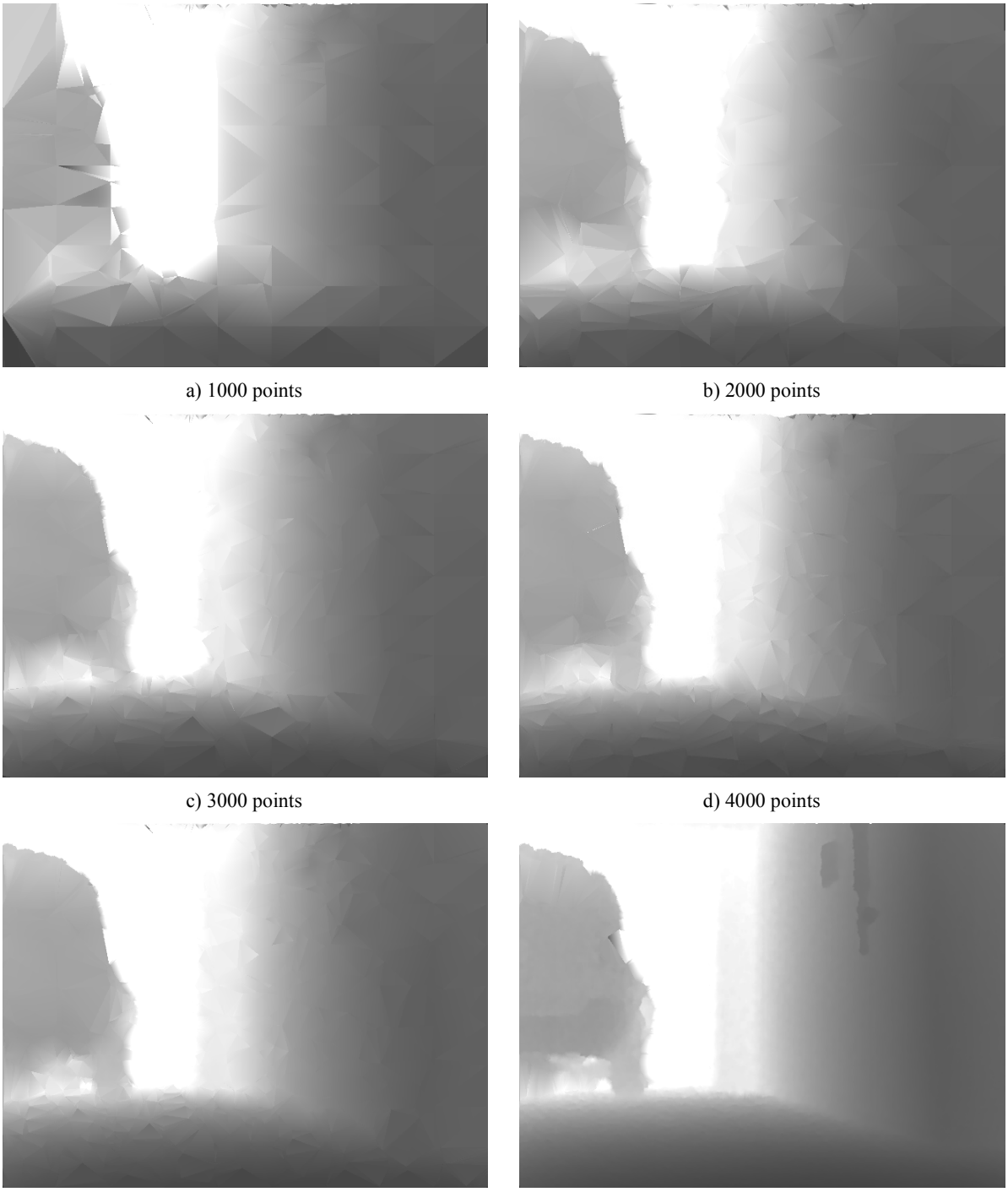
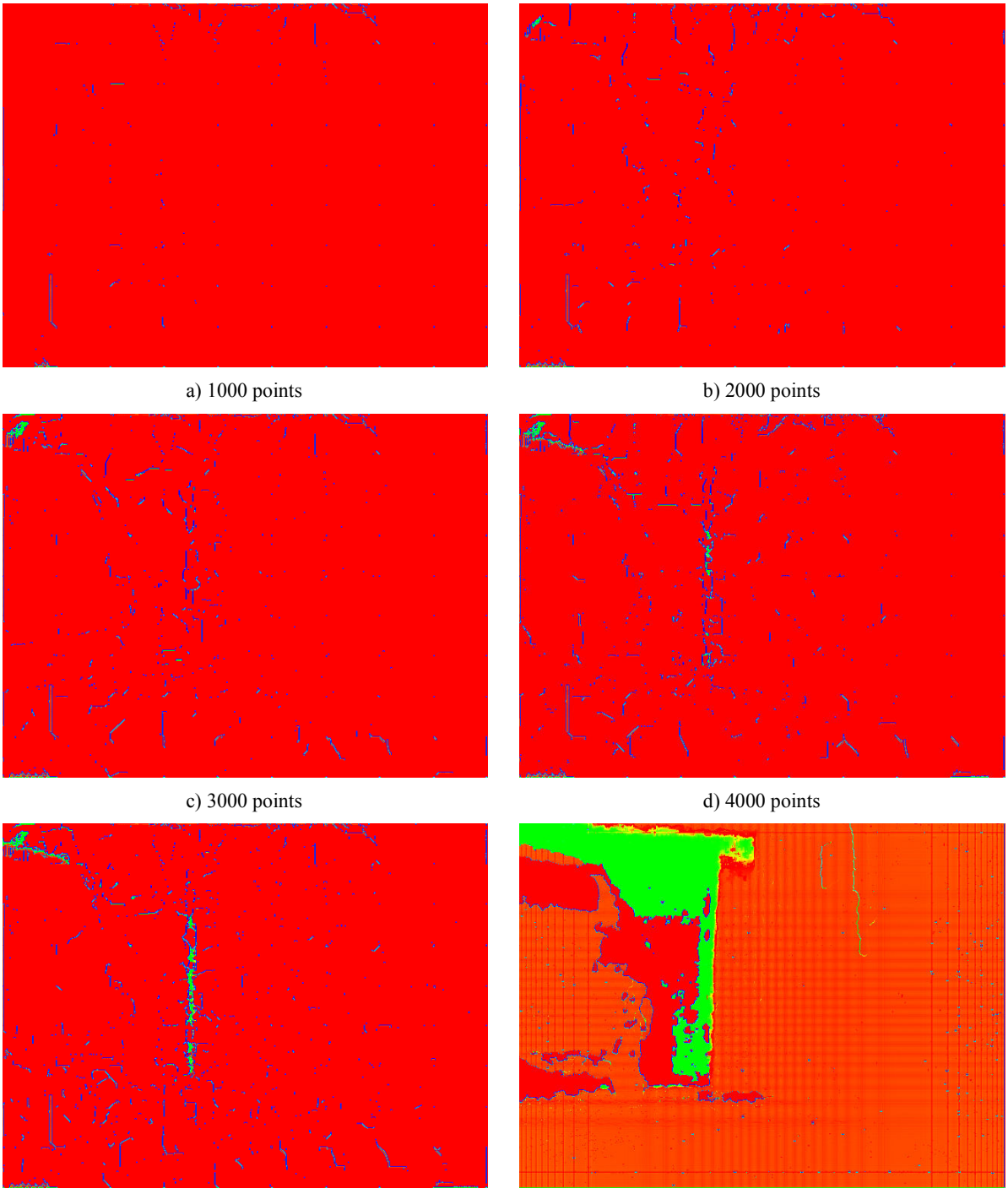


Figure D.8 - Scene 4: surface mesh based depth map progression after different numbers of points are acquired using the surface mesh improvement based selection process



a) 1000 points

b) 2000 points

c) 3000 points

d) 4000 points

e) 5000 points

f) all points

Figure D.9 - Scene 4: probabilistic occupancy grid improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

D.2.3 Scene 5 - Parking Garage Pay Machine

As can be observed for Scene 5 depicted in Section C.5, the progression of improvement in both Figure D.10 and Figure D.12 illustrates that the outlines of the pillars are acquired first (in subfigures a) b) and c)), as these are smaller objects that have larger depth transitions at their edges. The progressive acquisition of points, using the proposed surface mesh improvement based selection process, then focuses on capturing the boundaries of the features of the pay machine (subfigures c), d), and e)). In Figure D.10 b)-f) a large cluster of white triangles in the left hand part of the scene can be observed as consisting of a region of high improvement based on the surface mesh improvement measure. These triangles are located where the Kinect does not acquire any depth measurements, as is reflected by the residual high probabilistic occupancy improvement value (red) assigned to the same region in Figure D.12 f) after all available points are acquired. After 3000-4000 points the scene is resolved sufficiently such that all the major features are recognizable (Figure D.11 c) and d)). This corresponds to the convergence that can be observed from the surface mesh estimated improvement error of this scene presented in Figure D.3 d), and the near convergence of the surface mesh mean estimated improvement presented in Figure D.3 b).

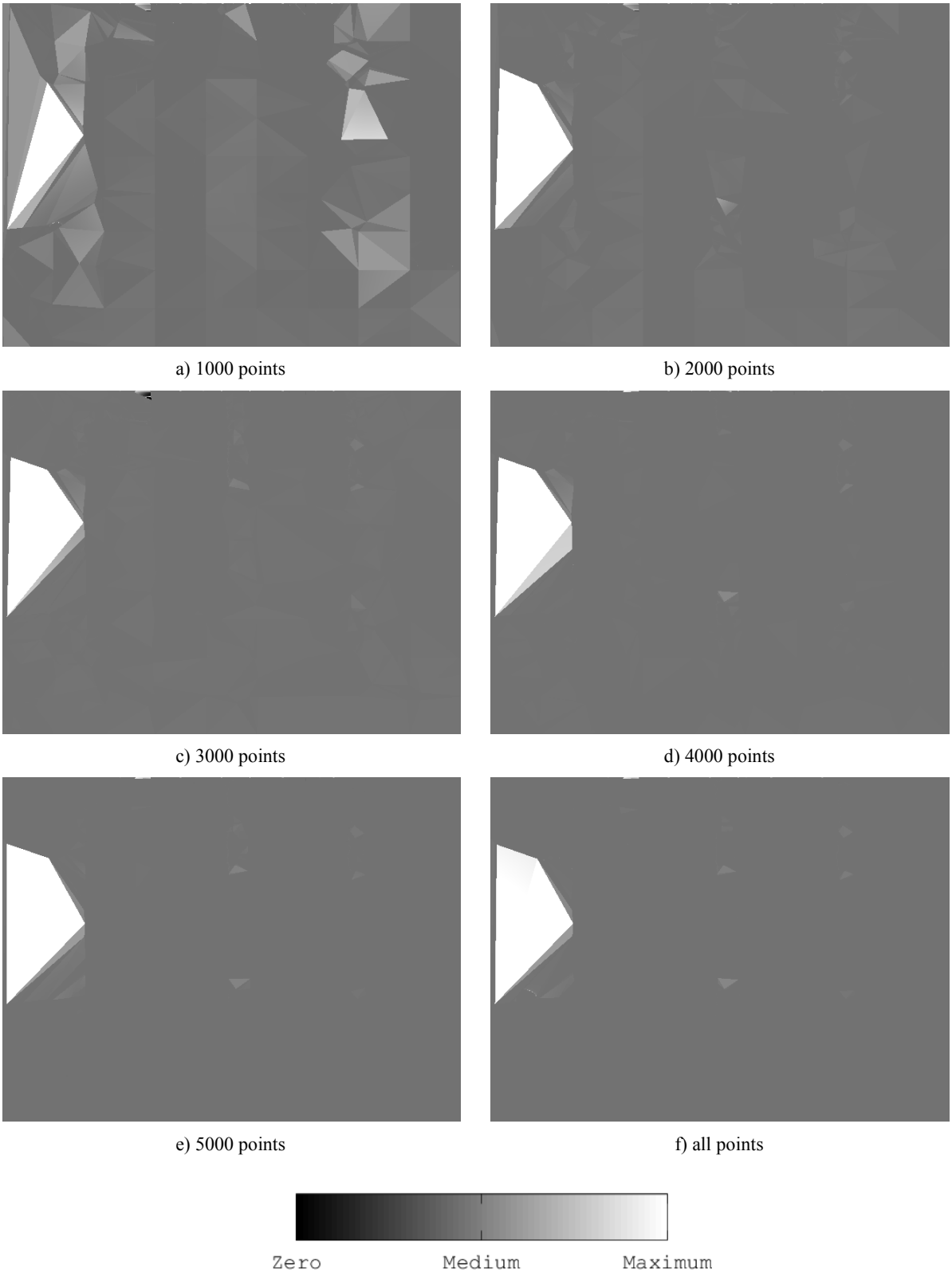


Figure D.10 - Scene 5: surface mesh improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

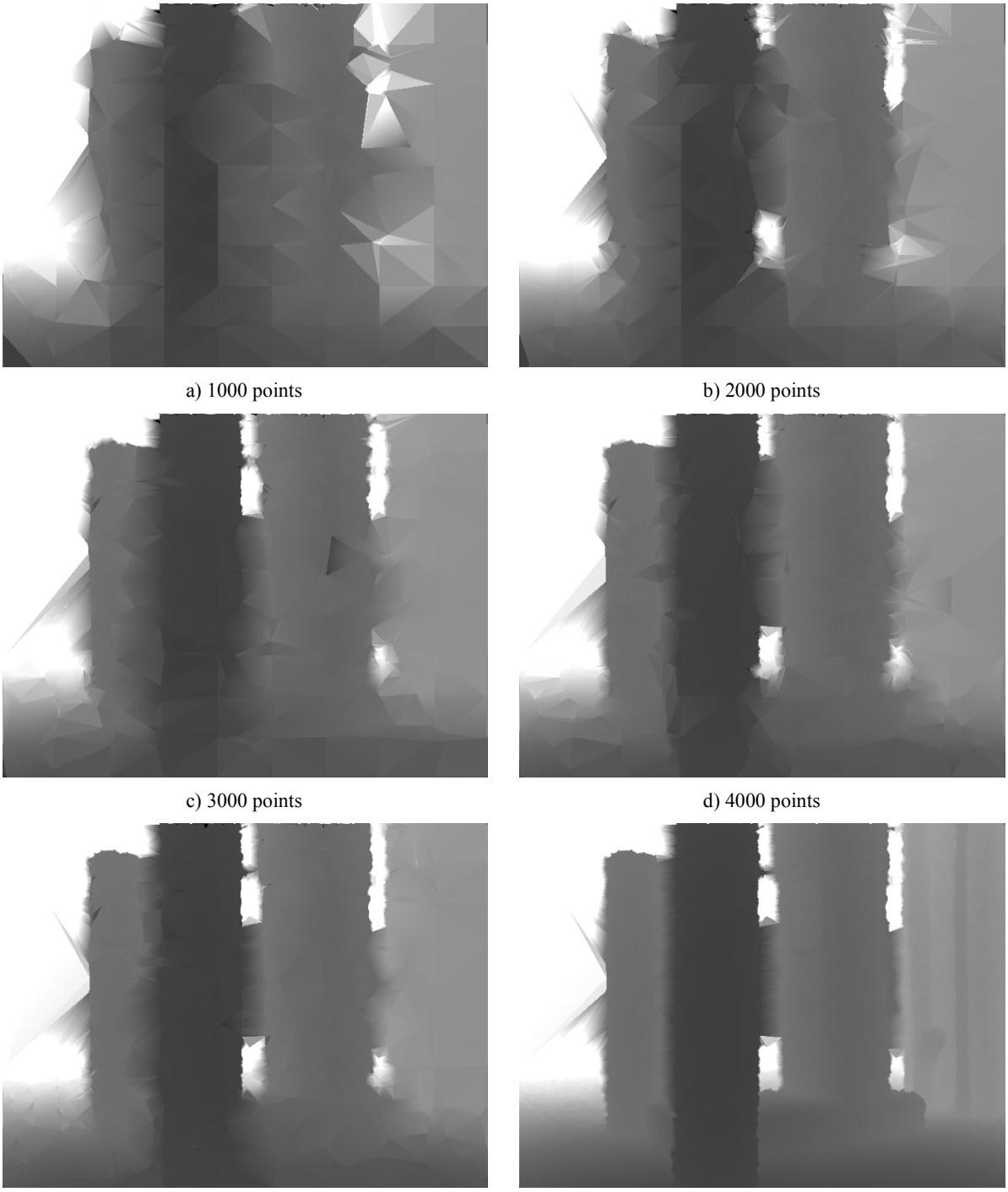
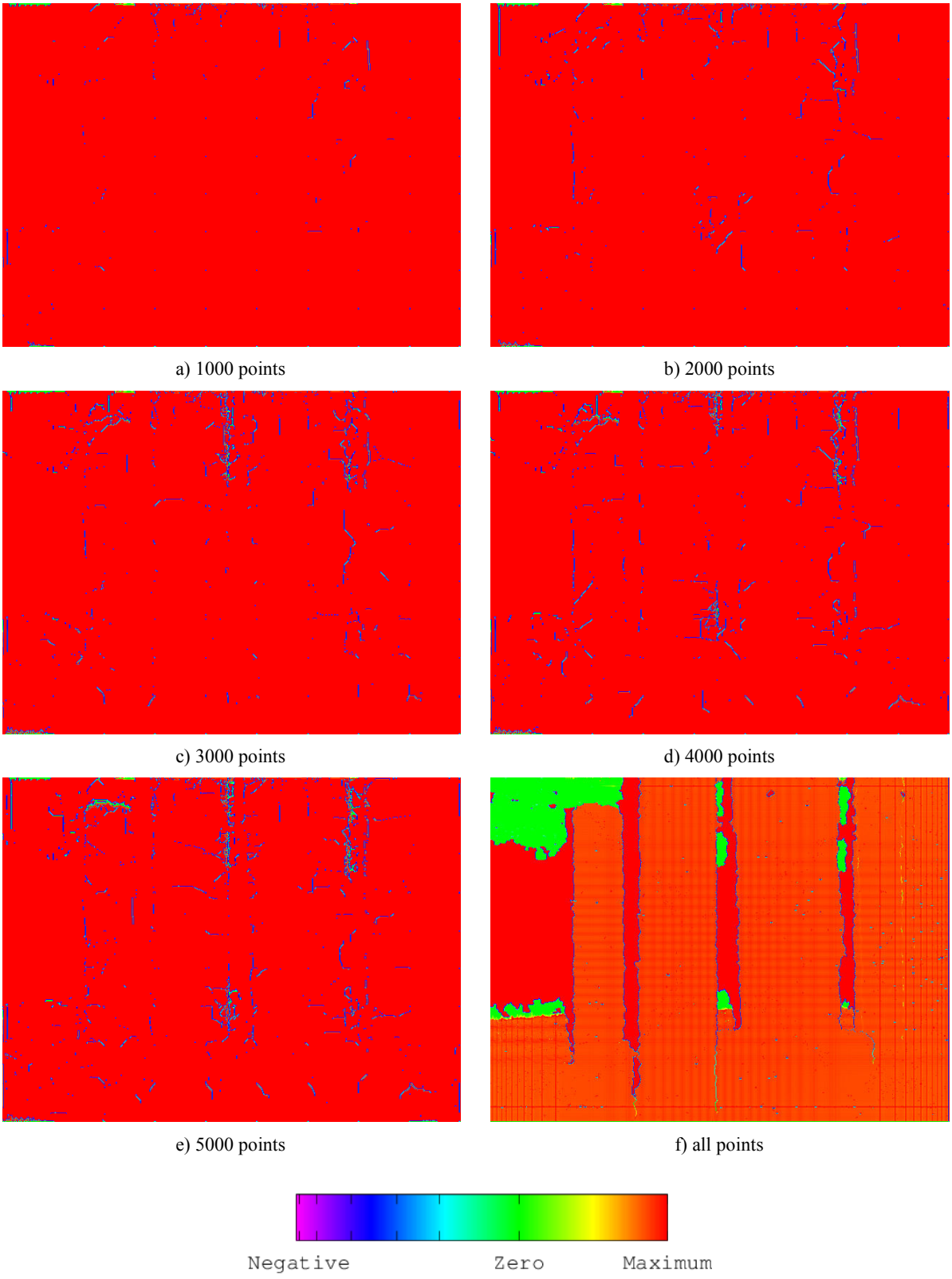


Figure D.11 - Scene 5: surface mesh based depth map progression after different numbers of points are acquired using the surface mesh improvement based selection process



a) 1000 points

b) 2000 points

c) 3000 points

d) 4000 points

e) 5000 points

f) all points

Figure D.12 - Scene 5: probabilistic occupancy grid improvement map progression after different numbers of points are acquired using the surface mesh improvement based selection process

D.3 Additional Execution Performance Results of the Surface Mesh Improvement Based Selection Process

This section provides additional execution performance results from the analysis presented in Section 5.2.3. The resulting performance for Scenes 3, 5, and 6 are presented.

Point Acquisition Method	Number of Points	SM Insertion Time (sec)	Mean SM Insertion Time Per Point (sec)	SM Improvement Calculation Time (sec)	SM Total Time (sec)	Percent Time Spent for SM Improvement Calculation over Total SM Time (%)	Mean SM Improvement Calculation Time Per Point (sec)	Point Selection Process Time (sec)	Mean Point Selection Time Per Point (sec)
SM DP(1)	276318	14.516	5.25E-05	90.604	105.12	86.191	0.000328	4.171	1.51E-05
SM DP(5)	276618	9.227	3.34E-05	75.796	85.022	89.148	0.000274	1.454	5.26E-06
SM DP(10)	276691	7.802	2.82E-05	67.096	74.897	89.583	0.000242	1.032	3.73E-06
SM DP(20)	276702	7.345	2.65E-05	66.173	73.518	90.009	0.000239	0.818	2.96E-06
Random (Mean)	307200	8.653	2.82E-05	53.767	62.420	86.137	0.000175	0.653	2.13E-06

Table D.1 - Scene 3: timing data for surface mesh improvement measure

Point Acquisition Method	Number of Points	POG Insertion Time (sec)	Mean POG Insertion Time Per Point (sec)	POG Improvement Calculation Time (sec)	POG Total Time (sec)	Percent Time Spent for POG Improvement Calculation over Total POG Time (%)	Mean POG Improvement Calculation Time Per Point (sec)
SM DP(1)	276318	36.257	1.31E-04	2.421	38.678	6.259	8.76E-06
SM DP(5)	276618	33.312	1.20E-04	1.757	35.069	5.01	6.35E-06
SM DP(10)	276691	32.182	1.16E-04	1.611	33.794	4.767	5.82E-06
SM DP(20)	276702	31.527	1.14E-04	1.54	33.067	4.657	5.57E-06
Random (Mean)	307200	37.096	1.21E-04	2.415	39.511	6.113	7.86E-06

Table D.2 - Scene 3: timing data for probabilistic occupancy grid improvement measure

Point Acquisition Method	Number of Points	SM Insertion Time (sec)	Mean SM Insertion Time Per Point (sec)	SM Improvement Calculation Time (sec)	SM Total Time (sec)	Percent Time Spent for SM Improvement Calculation over Total SM Time (%)	Mean SM Improvement Calculation Time Per Point (sec)	Point Selection Process Time (sec)	Mean Point Selection Time Per Point (sec)
SM DP(1)	237369	12.951	5.46E-05	86.234	99.185	86.943	0.000363	4.111	1.73E-05
SM DP(5)	237591	8.324	3.50E-05	69.639	77.964	89.323	0.000293	1.412	5.94E-06
SM DP(10)	237503	7.416	3.12E-05	68.295	75.711	90.205	0.000288	1.072	4.51E-06
SM DP(20)	237544	7.054	2.97E-05	66.06	73.115	90.352	0.000278	0.832	3.50E-06
Random (Mean)	259676	7.347	2.83E-05	49.299	56.646	87.03	0.00019	0.554	2.13E-06

Table D.3 - Scene 5: timing data for surface mesh improvement measure

Point Acquisition Method	Number of Points	POG Insertion Time (sec)	Mean POG Insertion Time Per Point (sec)	POG Improvement Calculation Time (sec)	POG Total Time (sec)	Percent Time Spent for POG Improvement Calculation over Total POG Time (%)	Mean POG Improvement Calculation Time Per Point (sec)
SM DP(1)	237369	26.625	1.12E-04	2.029	28.653	7.08	8.55E-06
SM DP(5)	237591	23.803	1.00E-04	1.541	25.344	6.079	6.48E-06
SM DP(10)	237503	23.348	9.83E-05	1.358	24.706	5.497	5.72E-06
SM DP(20)	237544	22.593	9.51E-05	1.396	23.989	5.82	5.88E-06
Random (Mean)	259676	27.201	1.05E-04	2.060	29.262	7.041	7.93E-06

Table D.4 - Scene 5: timing data for probabilistic occupancy grid improvement measure

Point Acquisition Method	Number of Points	SM Insertion Time (sec)	Mean SM Insertion Time Per Point (sec)	SM Improvement Calculation Time (sec)	SM Total Time (sec)	Percent Time Spent for SM Improvement Calculation over Total SM Time (%)	Mean SM Improvement Calculation Time Per Point (sec)	Point Selection Process Time (sec)	Mean Point Selection Time Per Point (sec)
SM DP(1)	251328	16.382	6.52E-05	141.762	158.14	89.642	0.000564	4.302	1.71E-05
SM DP(5)	251624	8.882	3.53E-05	76.644	85.525	89.615	0.000305	1.463	5.81E-06
SM DP(10)	251681	7.773	3.09E-05	69.445	77.218	89.934	0.000276	1.072	4.26E-06
SM DP(20)	251583	7.109	2.83E-05	67.084	74.194	90.418	0.000267	0.753	2.99E-06
Random (Mean)	277753	7.781	2.80E-05	52.887	60.668	87.175	0.00019	0.607	2.18E-06

Table D.5 - Scene 6: timing data for surface mesh improvement measure

Point Acquisition Method	Number of Points	POG Insertion Time (sec)	Mean POG Insertion Time Per Point (sec)	POG Improvement Calculation Time (sec)	POG Total Time (sec)	Percent Time Spent for POG Improvement Calculation over Total POG Time (%)	Mean POG Improvement Calculation Time Per Point (sec)
SM DP(1)	251328	29.336	1.17E-04	2.18	31.515	6.917	8.67E-06
SM DP(5)	251624	26.521	1.05E-04	1.632	28.153	5.797	6.49E-06
SM DP(10)	251681	26.093	1.04E-04	1.557	27.651	5.631	6.19E-06
SM DP(20)	251583	25.302	1.01E-04	1.538	26.84	5.73	6.11E-06
Random (Mean)	277753	30.557	1.10E-04	2.297	32.855	6.993	8.27E-06

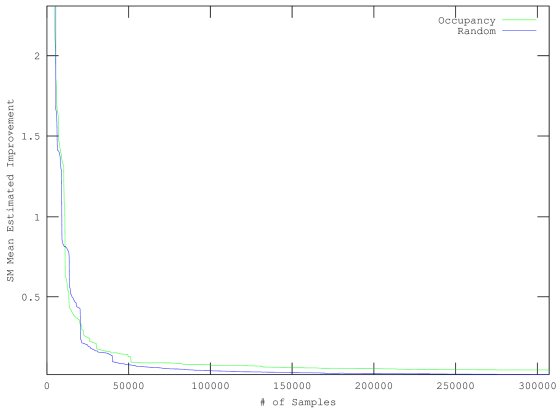
Table D.6 - Scene 6: timing data for probabilistic occupancy grid improvement measure

D.4 Additional Quantitative Analysis of the Probabilistic Occupancy Grid Improvement Based Selection Process

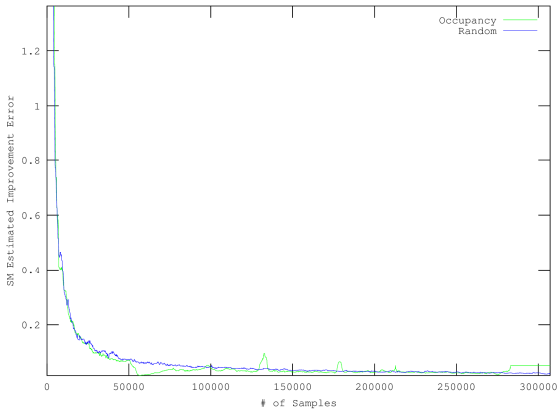
This section provides additional analysis of the six scenes that expand upon that presented in Section 5.3.1. The analysis of each scene is presented in its own subsection for easy indexing and access.

D.4.1 Analysis of Scene 2

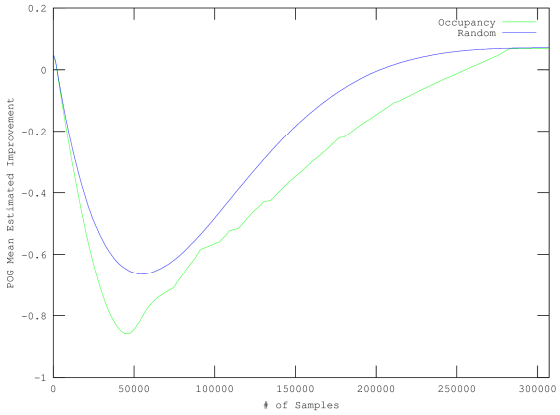
The second scene (see Section C.2) is an empty corner in a parking garage. Figure D.13 a) of the surface mesh mean estimated improvement metric illustrates that the curve for the random point selection process converges to the minimum slightly earlier than that of the probabilistic occupancy grid improvement based selection process, and hence is slightly better at capturing scene accuracy. The plot of the surface mesh estimated improvement error, Figure D.13 b), shows that the curve for the progressive occupancy grid improvement selection process decays earlier to the minimal value than does the curve for the random point selection process, but there is more noise present in this curve. Figure D.13 c), the plot of the probabilistic occupancy grid mean estimated improvement, indicates that the probabilistic occupancy grid improvement based selection process performs better at ensuring scene coverage earlier on in the acquisition process (with the minimum occurring at about 45000 points chosen) than does the random point selection process. The plot of the probabilistic occupancy grid estimated improvement error, Figure D.13 d), again shows that the probabilistic occupancy grid improvement based selection process is indeed better than the random point selection process at focussing acquisition of scene coverage, as the curve for the former is much lower value than the latter over the majority of the scene capture.



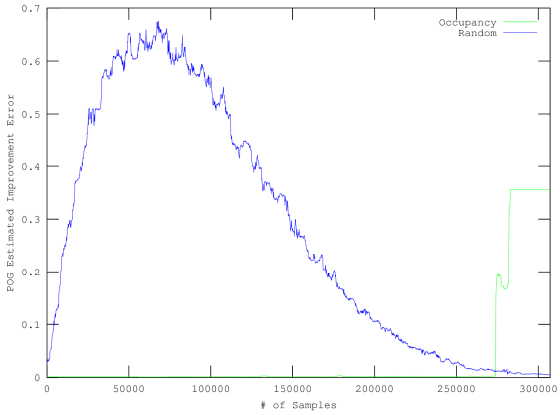
a) Surface mesh mean estimated improvement over all points



b) Surface mesh estimated improvement error over all points



c) Probabilistic occupancy grid mean estimated improvement over all points



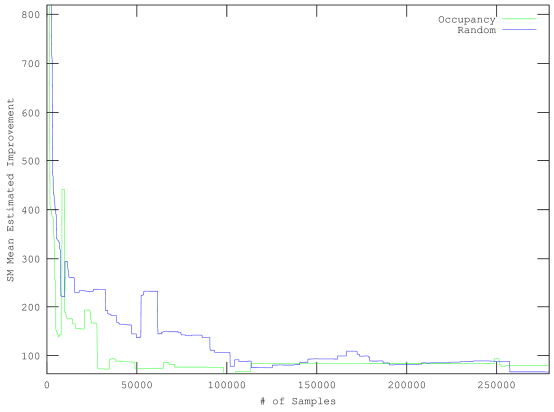
d) Probabilistic occupancy grid estimated improvement error over all points

Figure D.13 - Scene 2: mean improvement and improvement error trend graphs

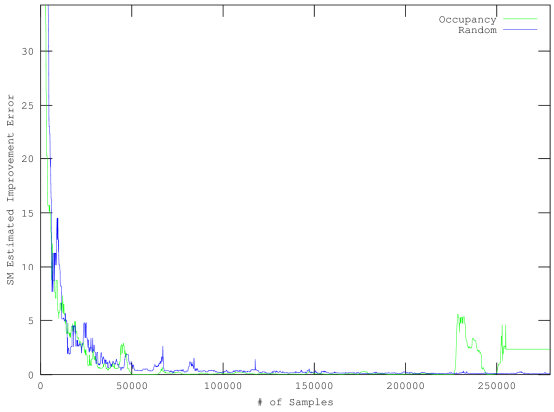
D.4.2 Analysis of Scene 4

Scene 4, shown in Section C.4, consists of a car beside a wall in a parking garage. The plot of the surface mesh mean estimated improvement, shown in Figure D.14 a), indicates that for this scene, the probabilistic occupancy grid improvement based selection process guides the acquisition to better accuracy significantly earlier than that performed by the random point selection process. This is likely to be a performance outlier due to the random nature of arbitrarily choosing one of the many locations which contain equal maximal probabilistic occupancy grid improvement values. Figure D.14 b), plot of the surface mesh estimated improvement error, shows that the curve representing the probabilistic occupancy grid improvement based selection process converges slightly early than does the curve representing the random point selection process, but it has a large spike in the error value at the end, likely due to the fact that depth transitions and edges are acquired last using the former process. The plot of the probabilistic occupancy grid mean estimated improvement,

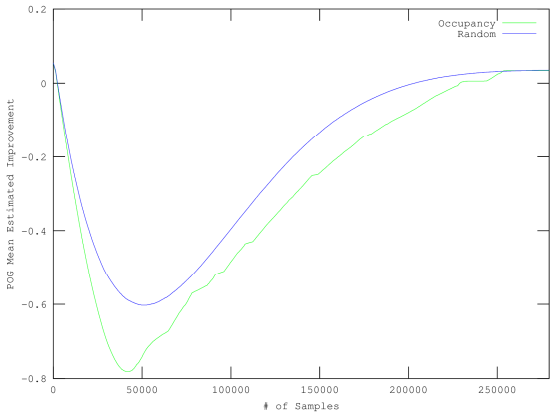
shown in Figure D.14 c), shows that the probabilistic occupancy grid improvement based selection process has better scene coverage, with the minimal value occurring after about 40000 points have be acquired, compared to the random point selection process. The plot of the probabilistic occupancy grid estimated improvement error in Figure D.14 d) also shows that the curve representing the probabilistic occupancy grid improvement based selection process is lower than that produced by the random point selection process, and hence the former technique is better at providing scene coverage.



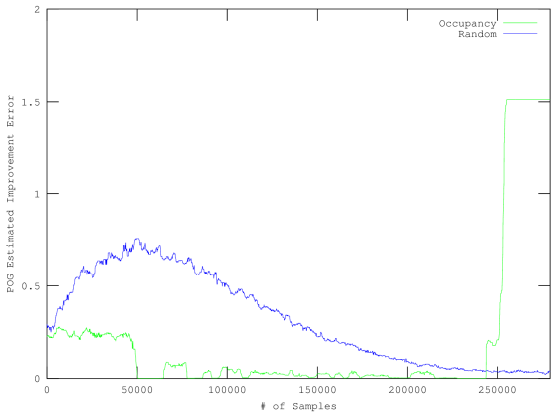
a) Surface mesh mean estimated improvement over all points



b) Surface mesh estimated improvement error over all points



c) Probabilistic occupancy grid mean estimated improvement over all points



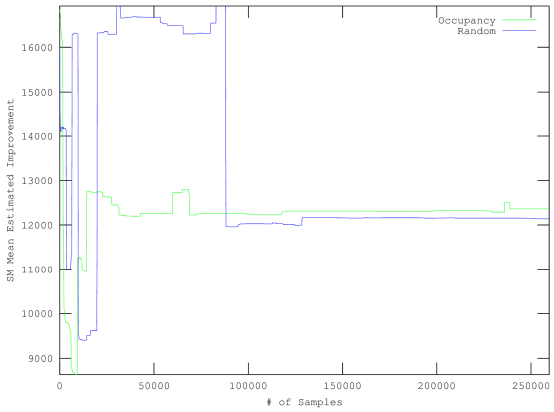
d) Probabilistic occupancy grid estimated improvement error over all points

Figure D.14 - Scene 4: mean improvement and improvement error trend graphs

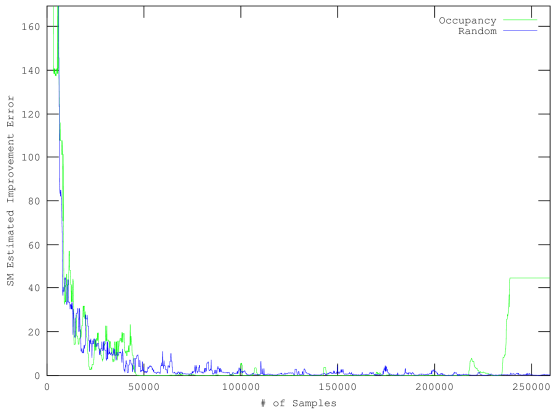
D.4.3 Analysis of Scene 5

Scene 5, shown in Section C.5, shows some pillars surrounding a pay machine within a parking garage. The plot of the surface mesh mean estimate improvement metric, Figure D.15 a) indicates for this scene that neither selection process performs well at ensuring early scene accuracy, as both curves exhibit much variation. Figure D.15 b) that contains the plot

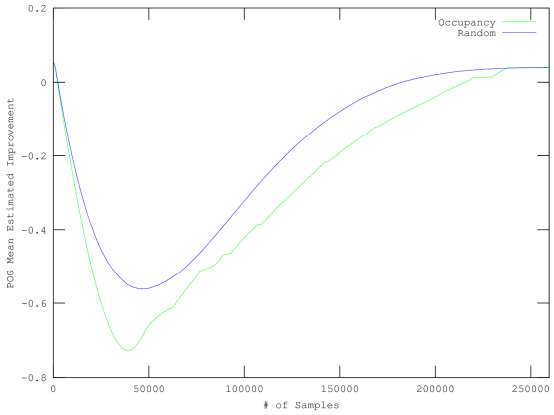
of surface mesh estimated improvement error, reinforces this interpretation, as the curves generated using both selection processes converge as similar rates, but the error as produced by probabilistic occupancy grid improvement based selection process, as observed in the previous scenes, contains a spike in this metric at the end of the acquisition process. This is likely due to edges and depth transition being acquired only at the end of the scene capture. The plot of the probabilistic occupancy grid mean estimated improvement, in Figure D.15 c), indicates that the probabilistic occupancy grid improvement based selection process does indeed provide better scene coverage (with the plot minimum occurring at about the 40000 points acquired mark) than the random point selection process. Figure D.15 d) showing the plot of the probabilistic occupancy grid estimate improvement error, illustrates that the curve representing the probabilistic occupancy grid improvement based selection process is lower for most of the acquisition sequence than the curve generated by the random point selection process, and hence the former selection process produces better scene coverage.



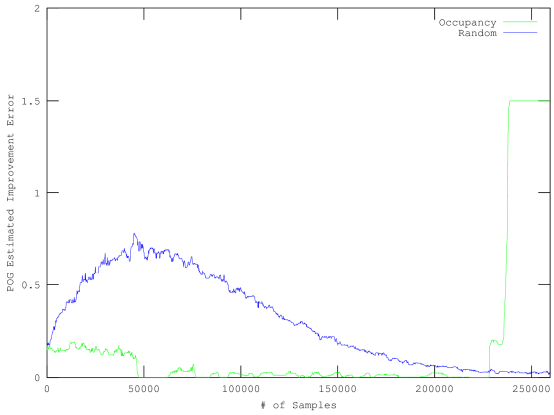
a) Surface mesh mean estimated improvement over all points



b) Surface mesh estimated improvement error over all points



c) Probabilistic occupancy grid mean estimated improvement over all points



d) Probabilistic occupancy grid estimated improvement error over all points

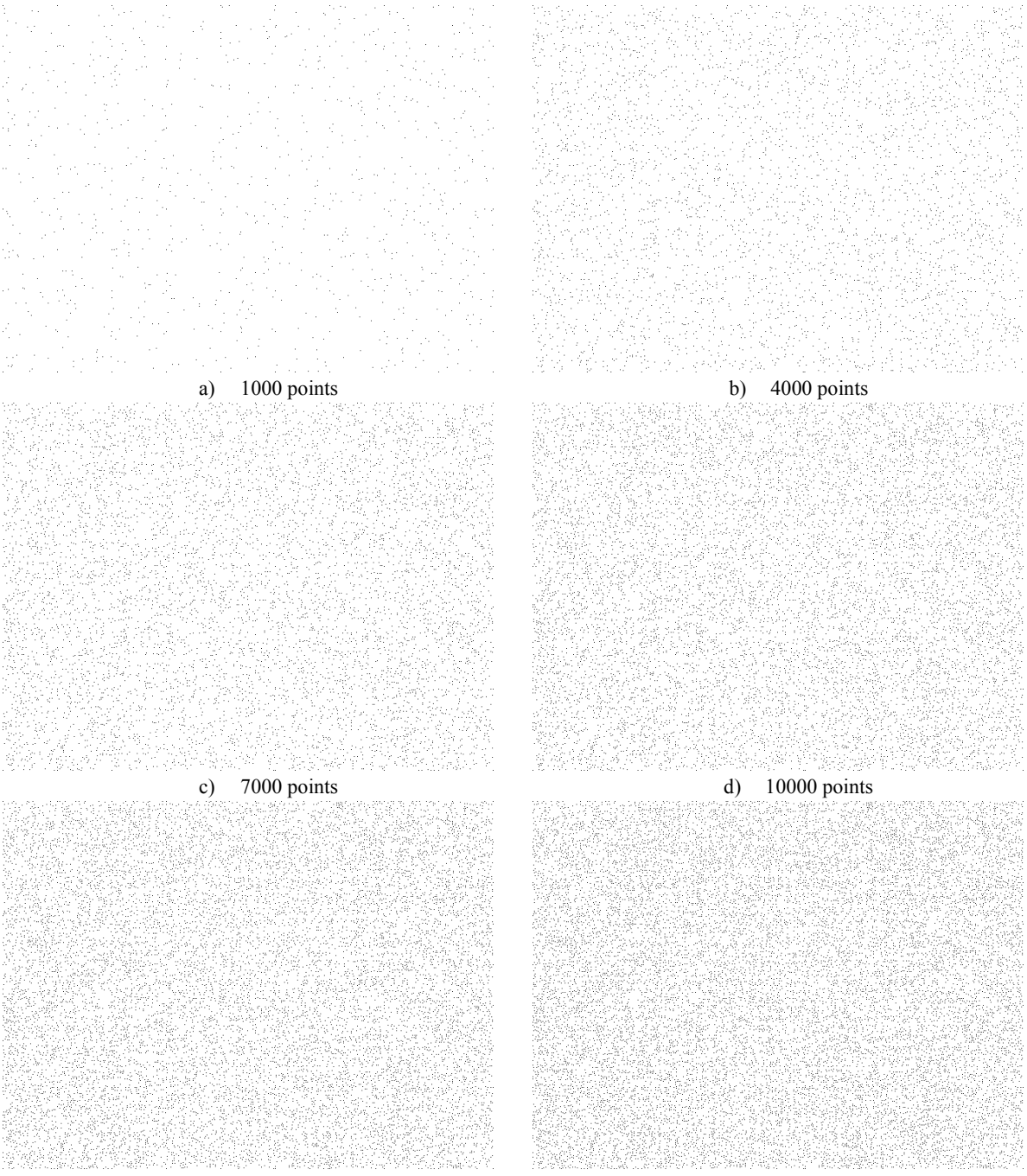
Figure D.15 - Scene 5: mean improvement and improvement error trend graphs

D.5 Additional Progression Results and Analysis of the Probabilistic Occupancy Grid Improvement Based Selection Process

The results demonstrating and analysing the progression for Scenes 2, 4, and 5 are presented in this section, as they are very much similar to the three other scenes presented in Section 5.3.2.

D.5.1 Scene 2: Corner in a Parking Garage

In Scene 2 (shown in Section C.2), of the empty parking garage corner, the progression of points chosen to be inserted by the probabilistic occupancy grid improvement based selection process is shown in Figure D.16. Notice that in the progression from 1000 points in Figure D.16 a), to 16000 points in Figure D.16 f), although the points appear to be distributed uniformly random, not a single point has been chosen that is adjacent to a location that has already been acquired. This is a result of the proposed probabilistic occupancy grid improvement based selection process being devoted to scene coverage. Figure D.17, which illustrates the probabilistic occupancy grid model for $P(Occ) > 0.5$, clearly shows that the model of the scene has been captured with the sole priority of coverage, with much of the scene being visible by the time 4000 points (Figure D.17 b)) have been integrated into the model. As the progression continues to 16000 points in Figure D.17 f), the surfaces of the scene become more densely covered with voxels. Although this representation appears dense, there are still gaps between voxels along the surface, which the probabilistic occupancy grid by itself cannot intrinsically interpolate the location.

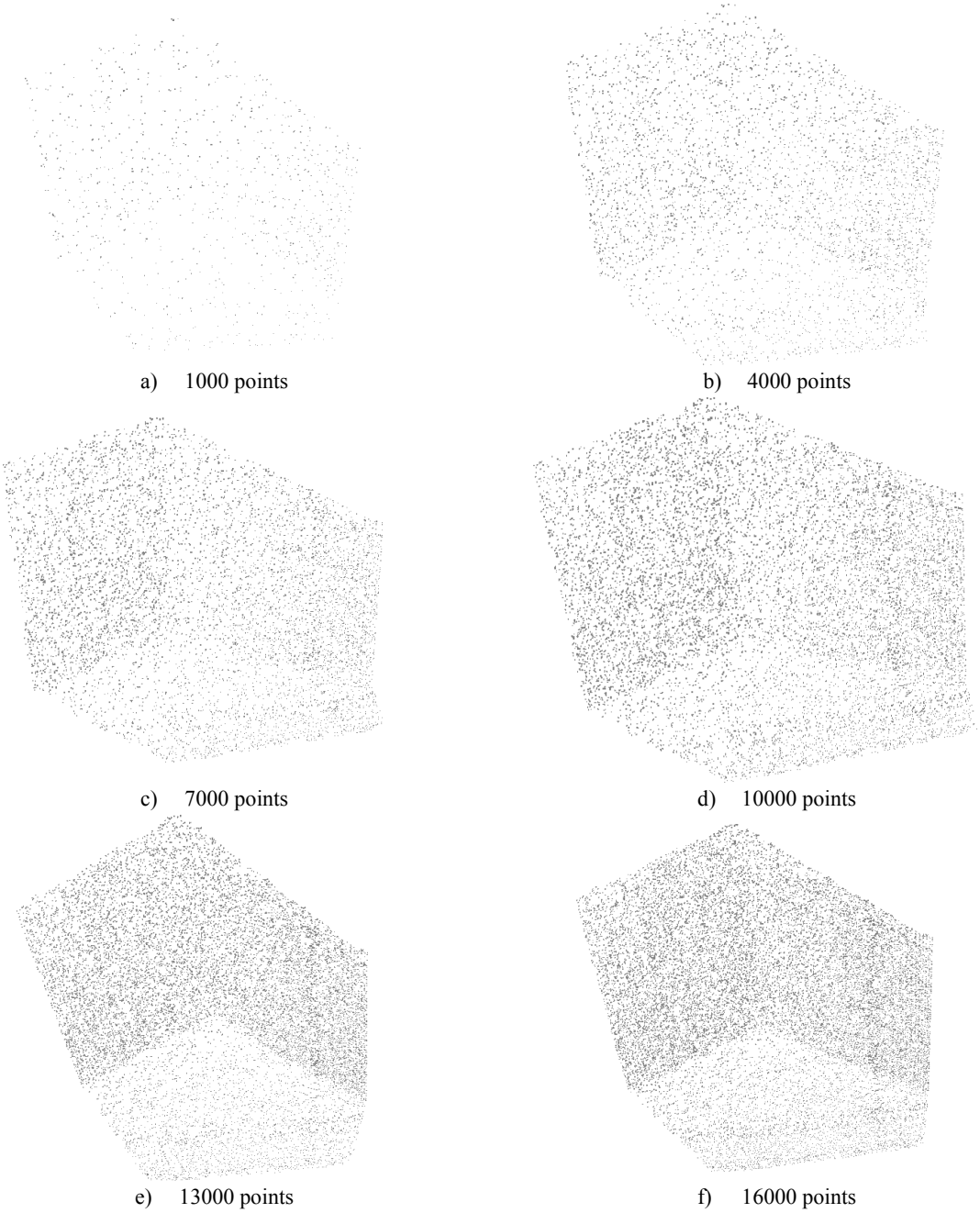


a) 1000 points b) 4000 points

c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

Figure D.16 - Scene 2: point selection map progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process



a) 1000 points

b) 4000 points

c) 7000 points

d) 10000 points

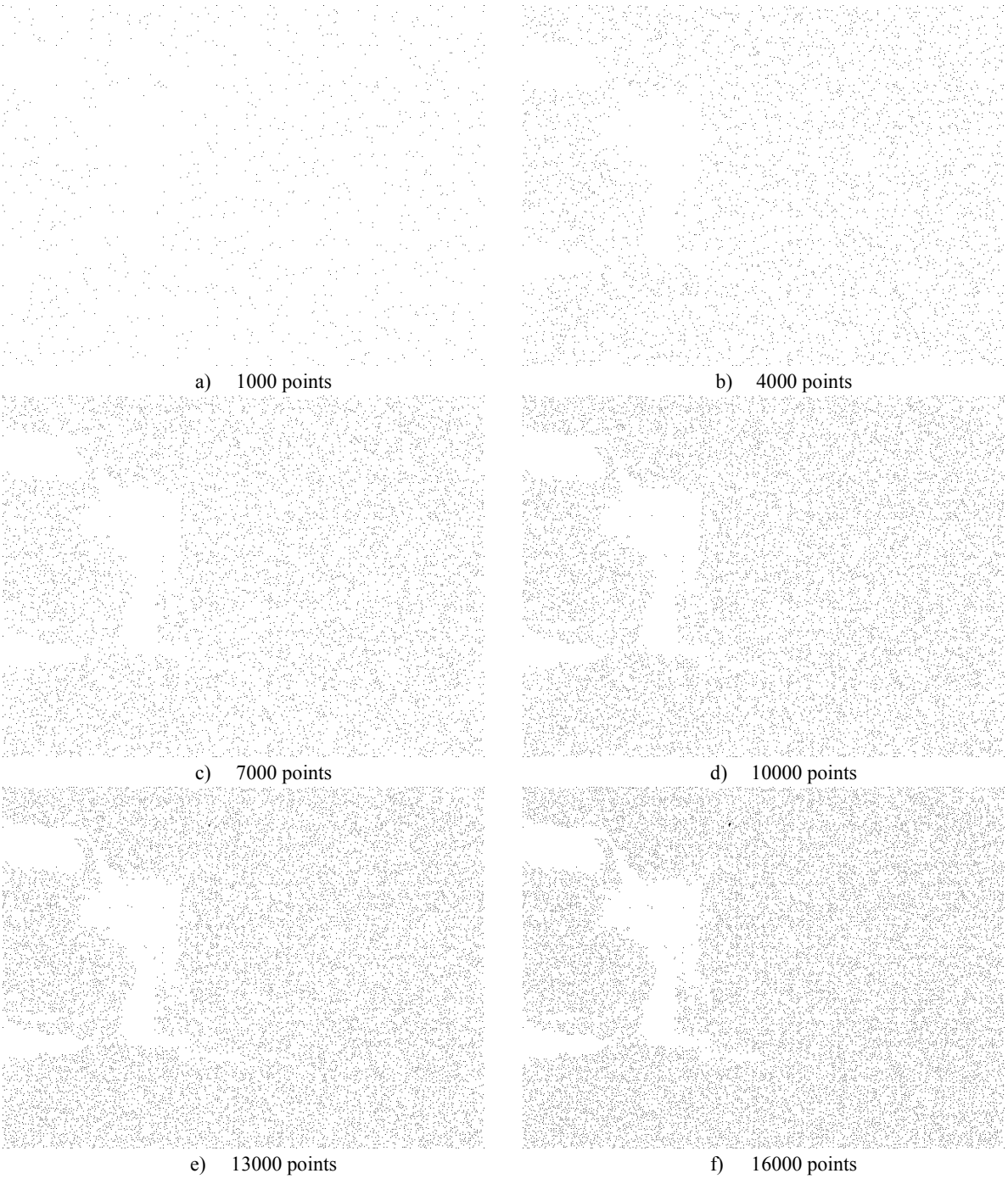
e) 13000 points

f) 16000 points

Figure D.17 - Scene 2: probabilistic occupancy grid model (for P(Occ)>0.5) progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process

D.5.2 Scene 4: Car and Walls in a Parking Garage

The progression of locations acquired in Scene 4, consisting of the car beside a wall in a parking garage (Section C.4), is shown in Figure D.18. Notice that there are no points chosen and inserted into the model in the regions where the Kinect cannot perceive, such as the regions where the car window, reflectors, and underneath the car are located. Furthermore, as the number of points chosen and inserted into the model increases from 1000 points in Figure D.18 a) to 16000 points in Figure D.18 f), the point density is fairly uniform at each stage, and there are no directly adjacent locations selected. This verifies that the probabilistic occupancy grid improvement based selection coverage does indeed direct acquisitions to provide better scene coverage. Figure D.19 details the progression of the probabilistic occupancy grid model for $P(Occ) > 0.5$ as points are selectively chosen and integrated into the model. The relative voxel densities on the surfaces of the car, ground, and walls are similar, regardless of the number of points selectively chosen and inserted into the model. This further verifies that the probabilistic occupancy grid improvement based selection process directs acquisitions to improve scene coverage. The voxels along the surfaces of the objects are fairly dense, but there is still much unknown space between them, that cannot be represented with this number of inserted points by the probabilistic occupancy grid.

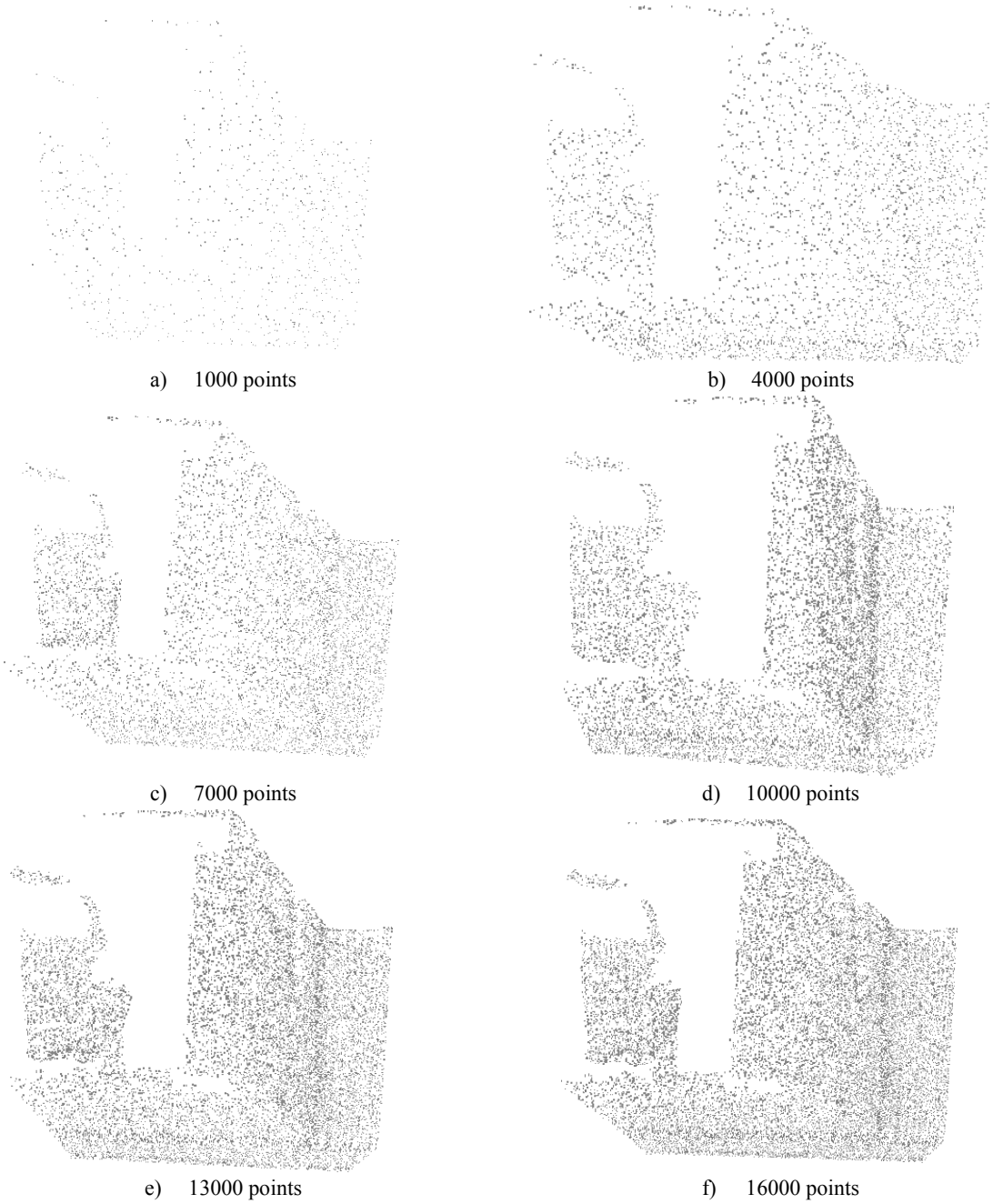


a) 1000 points b) 4000 points

c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

Figure D.18 - Scene 4: point selection map progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process



a) 1000 points

b) 4000 points

c) 7000 points

d) 10000 points

e) 13000 points

f) 16000 points

Figure D.19 - Scene 4: probabilistic occupancy grid model (for P(Occ) >0.5) progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process

D.5.3 Scene 5: Pay Machine and Pillars in a Parking Garage

As can be observed for Scene 5 depicted in Section C.5, the progression of the point locations selectively acquired and inserted into the model, shown in Figure D.20, illustrates that the objects (pillars and pay machine) as well as regions that the Kinect was able to acquire depth information on but that were outside of the supported region (up to 5.0m) are of uniform point density, regardless of how many points were selectively inserted, and that there are no immediately adjacent points chosen. This verifies the quantitative analysis made in Section 5.3.1, that the probabilistic occupancy grid improvement based selection process favours acquiring points to further improve scene coverage. The progression of the probabilistic occupancy grid model for $P(Occ) > 0.5$, shown in Figure D.21, demonstrates that the three pillars, and the pay machine, have similar voxel densities along their surfaces, regardless of the number of points selectively inserted into the model. Note that after 16000 points have been selectively inserted, Figure D.21 f), there are still plenty of unknown voxels along the various object surfaces that cannot be interpolated intrinsically by the probabilistic occupancy grid model.

a) 1000 points b) 4000 points

c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

Figure D.20 - Scene 5: point selection map progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process

a) 1000 points b) 4000 points

c) 7000 points d) 10000 points

e) 13000 points f) 16000 points

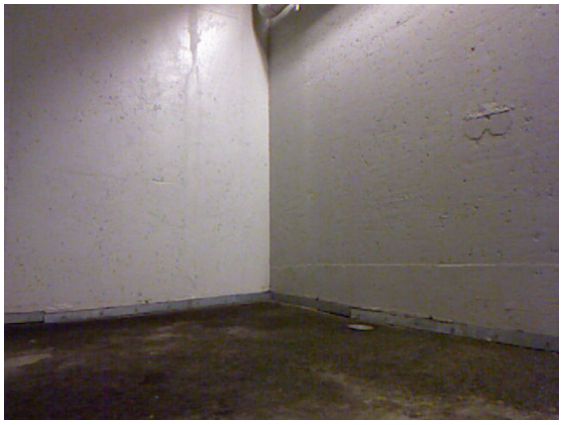
Figure D.21 - Scene 5: probabilistic occupancy grid model (for P(Occ)>0.5) progression after different numbers of points are acquired using the probabilistic occupancy grid improvement based selection process

D.6 Additional Analyses for the Exploration of Additional Applications of the Probabilistic Occupancy Grid Improvement Based Selection Process

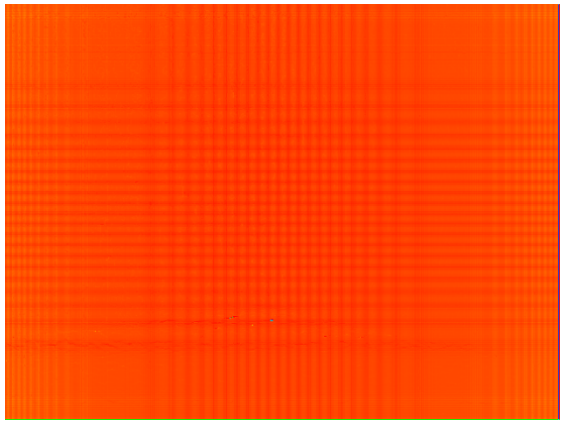
This section contains additional results from Section 5.3.3 detailing possible application of the probabilistic occupancy grid improvement map that occur after all available points for a scene are inserted. The result of this procedure, for Scenes 2, 3, and 5, are presented in the subsections below.

D.6.1 Scene 2 - Empty Parking Garage Corner

Figure D.22 c) is completely black, as all points in the scene have been acquired. Figure D.22 d), is black except for one point where the two walls and the ground intersect, as this is the only significant transition region in the image that causes negative improvement to be present. The distinctive thatch pattern appearing in Figure D.22 b) is a result of the non-linear mapping between points acquired by the Kinect in Cartesian coordinates, and the angular projection coordinates of the occupancy map.

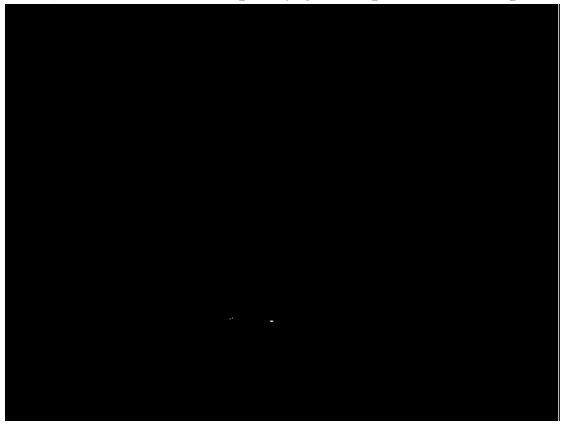


a) RGB image of scene



b) Probabilistic occupancy grid improvement map

c) High improvement regions (DP'_2)



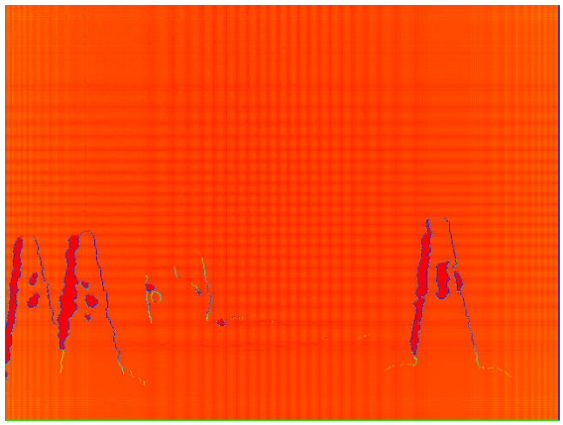
d) Negative improvement regions (DP''_2)

Figure D.22 - Probabilistic occupancy grid improvement based selection processes on Scene 2

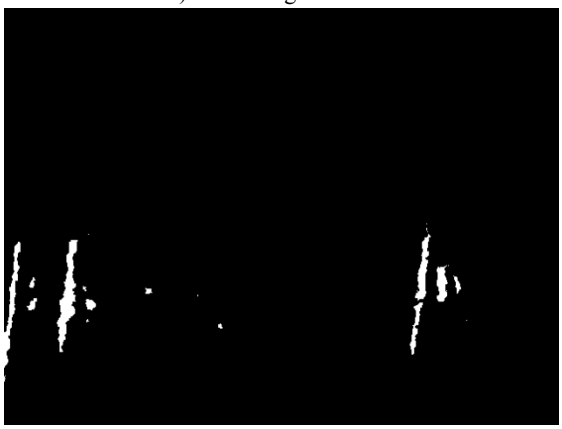
D.6.2 Scene 3 - Cluttered Corner in a Parking Garage

Scene 3 is of the same corner presented in Scene 2, but with some traffic cones and a filled recycling bin, as shown in Figure D.23 a). In Figure D.23, the regions of highest potential improvement correspond to where the IR reflects off of the shiny tape on the cones and the regions encompassing the occlusions, which are present due to the Kinect's IR projector and IR camera separation, around the cones (Figure D.23 c)). The negative improvement regions, caused by the depth transitions within the scene, outline the cones and parts of the recycling bin (Figure D.23 d)).

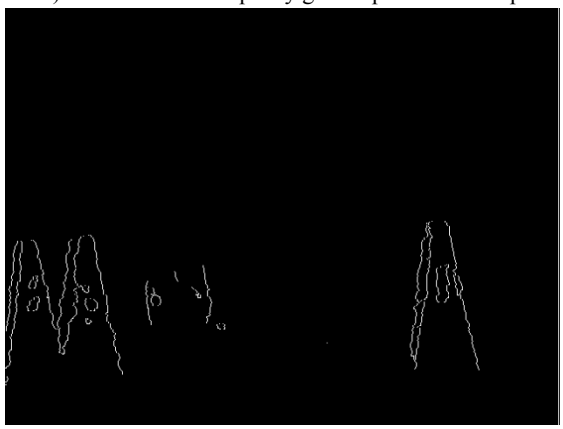
a) RGB image of scene



b) Probabilistic occupancy grid improvement map



c) High improvement regions (DP'_2)

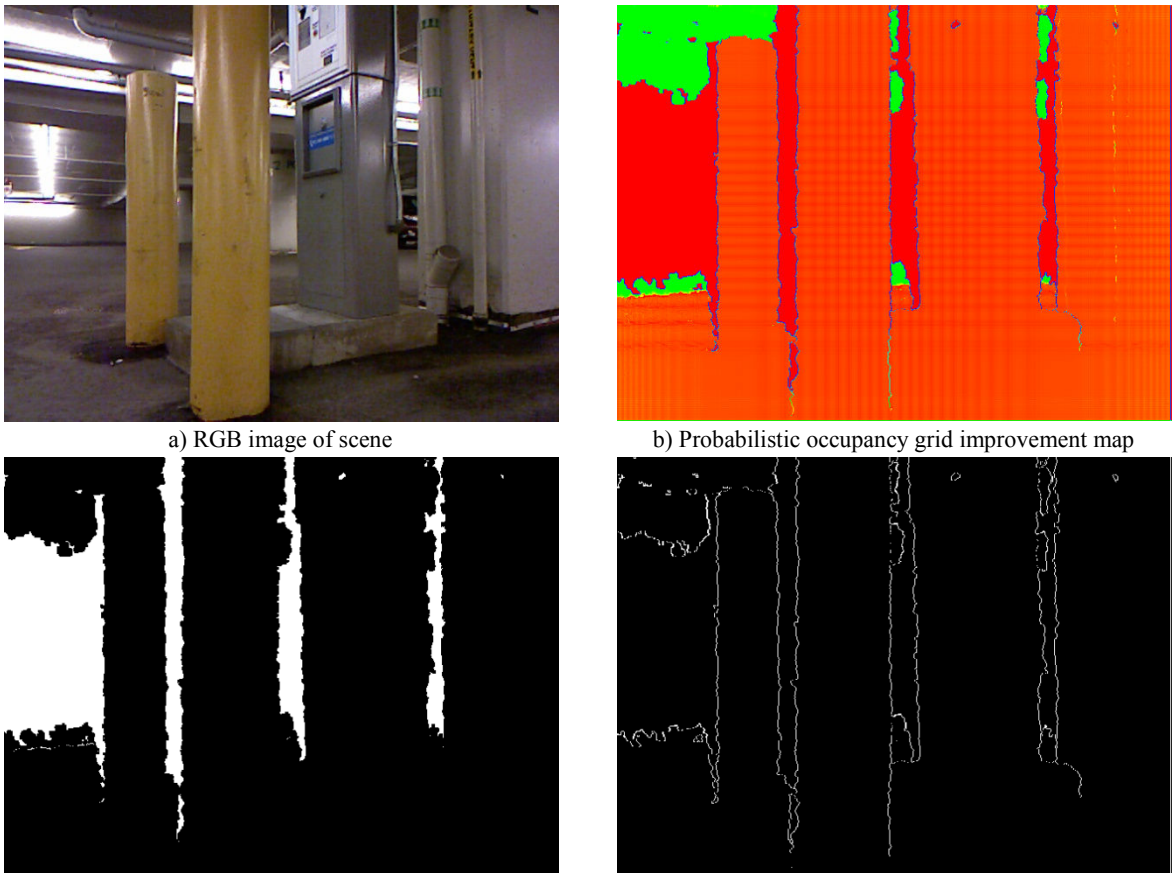


d) Negative improvement regions (DP''_2)

Figure D.23 - Probabilistic occupancy grid improvement based selection processes on Scene 3

D.6.3 Scene 5 - Pay Machine and Pillars in a Parking Garage

Figure D.24 a) shows Scene 5 of the pillars and a pay station in the parking garage. The negative improvement map (Figure D.24 d)) outlines the edges of the pillars and the pay station in the scene, as well the boundaries between unacquired space (red) and regions that were acquired but beyond the depth supported by the probabilistic occupancy grid (5.0m). The highest improvement, as highlighted in Figure D.24 c), lies in regions where there are no surfaces that the Kinect can acquire due to the depth being beyond the maximum depth of the sensor (on the left), and regions where there are occlusions produced by the separation of the IR projector and camera of the Kinect (partly in the regions between pillars).



a) RGB image of scene

b) Probabilistic occupancy grid improvement map

c) High improvement regions (DP'_2)

d) Negative improvement regions (DP''_2)

Figure D.24 - Probabilistic occupancy grid improvement based selection processes on Scene 5

D.7 Additional Comparative Progression Analysis for the Combined Improvement Selection Process

The following sections provide the complementary comparative progression analysis of Scenes 11 and 13, to what is presented in Section 5.4.2.

D.7.1 Scene 11: Computer

Figure D.25 illustrates the progressive selection of locations by the selection processes on the computer scene (Scene 11) as acquired by the Kinect. As can be observed after 2000 points (Figure D.25 a)), the combined improvement selection process (red) has the edges of the computer better covered than the surface mesh improvement based selection process (green). By 5000 points, both selection processes have densely chosen points along the sharpest depth transitions (indicated by the black pixels that indicate that more than one selection process has acquired those particular locations), and additionally the keyboard, as well as the other edges have started to be chosen by the combined improvement based selection process. Both the probabilistic occupancy grid improvement based selection process (magenta) and the random point selection process (blue) continue to provide better coverage of the scene in both cases by acquiring locations that are uniformly distributed within the scene.

The progressive choosing of locations by the selection process of the same scene as acquired by the ASLS, Figure D.26, details that both the combined improvement selection process (red) and the surface mesh selection process (green) have chosen, initially after 2000 points, the regions around the keyboard and the chassis, with some locations chosen on the monitor. By the 5000^{th} point, the combined improvement selection process has captured points along the surface of the chassis that include such features as the drive bay, plastic detailing, and the edge where the chassis sits on the covered platform, while the surface mesh improvement selection process has selected many of the same features, but with less coverage than the former technique. The probabilistic occupancy grid improvement based selection process (magenta) exhibits the same clustering behaviour as described for the covered exercise ball scene in Section 5.4.2.1. The random point selection process (blue) provides a uniform distributed set of point locations choices after both 2000 and 5000 points acquired, with lower density over the monitor as the sensor did not acquire this region successfully, as shown in Figure C.11 c).

Figure D.27 details the comparison of point locations chosen by the four selection processes, over the progression of a) 1000 points, b) 3000 points, and c) 5000 points, for the computer scene as acquired by the Jupiter mounted on the F3T robotic arm. As can be observed from Figure D.27 a), after 1000 points chosen the combined improvement selection process (red) has acquired points along the interior edge of the monitor, as well as the external edges of the scene, while the surface mesh improvement based selection process (green) provides a more directed acquisition along the surfaces connected to these edges. As the progression moves on to having 3000 and 5000 points selected, both of these selection processes chose more points where there are larger depth transitions in the scene (along the four corners and bottom of the maps presented in Figure D.27), with the combined selection process choosing more points with better coverage along the glass surface of the monitor (central region of the maps). The probabilistic occupancy grid improvement based selection process (magenta) has chosen points that are connected more or less horizontally, and that are distributed uniformly throughout the sensor's field of view, while the random point selection process (blue) has chosen points that are uniformly distributed.

Figure D.28 illustrates the comparison of the progression of point locations chosen by the four selection processes on the computer scene as acquired by the Jupiter mounted on the linear motorized track. As observed from Figure D.28 a), both the combined improvement process (red) and the surface mesh improvement based selection process (green) have chosen locations along the edge of the chassis, but they have focused much of their attention on acquiring the lines of noise present in the scene. By the time 5000 points have been acquired (Figure D.28 b)), the combined improvement process has chosen many points along the edge of the monitor, and along the keyboard, while the surface mesh improvement based selection process has only started acquiring locations along the keyboard. The random point selection process (blue) has chosen points uniformly distributed within the scene, while the probabilistic occupancy grid improvement based selection process (magenta) has chosen points that are nearly horizontally adjacent to each other, but these 'dotted' horizontal lines are uniformly distributed. These trends continue between 2000 and 5000 points selectively inserted into the model.

The progression of point locations acquired by the four selection processes for the computer scene as acquired by the SICK LMS is presented in Figure D.29. Note that as previously discussed, the vertical axis was sampled with a lower sampling rate and covered a larger field of view than the horizontal axis, hence the scene appears to be vertically compressed, with the computer occupying the middle vertical third of the captured scene. As presented through the progression of the red pixels, representing the combined improvement selection process, compared to the green pixels, representing the surface mesh improvement based selection process, the former process chooses locations along the top of the chassis (Figure D.29 b)) and the top of the monitor (Figure D.29 c)) prior to the latter process. The other two selection processes, the random point selection process (blue) and the probabilistic occupancy grid improvement based selection process (magenta) choose points that uniform cover the scene for each of the three cases studied in Figure D.29.

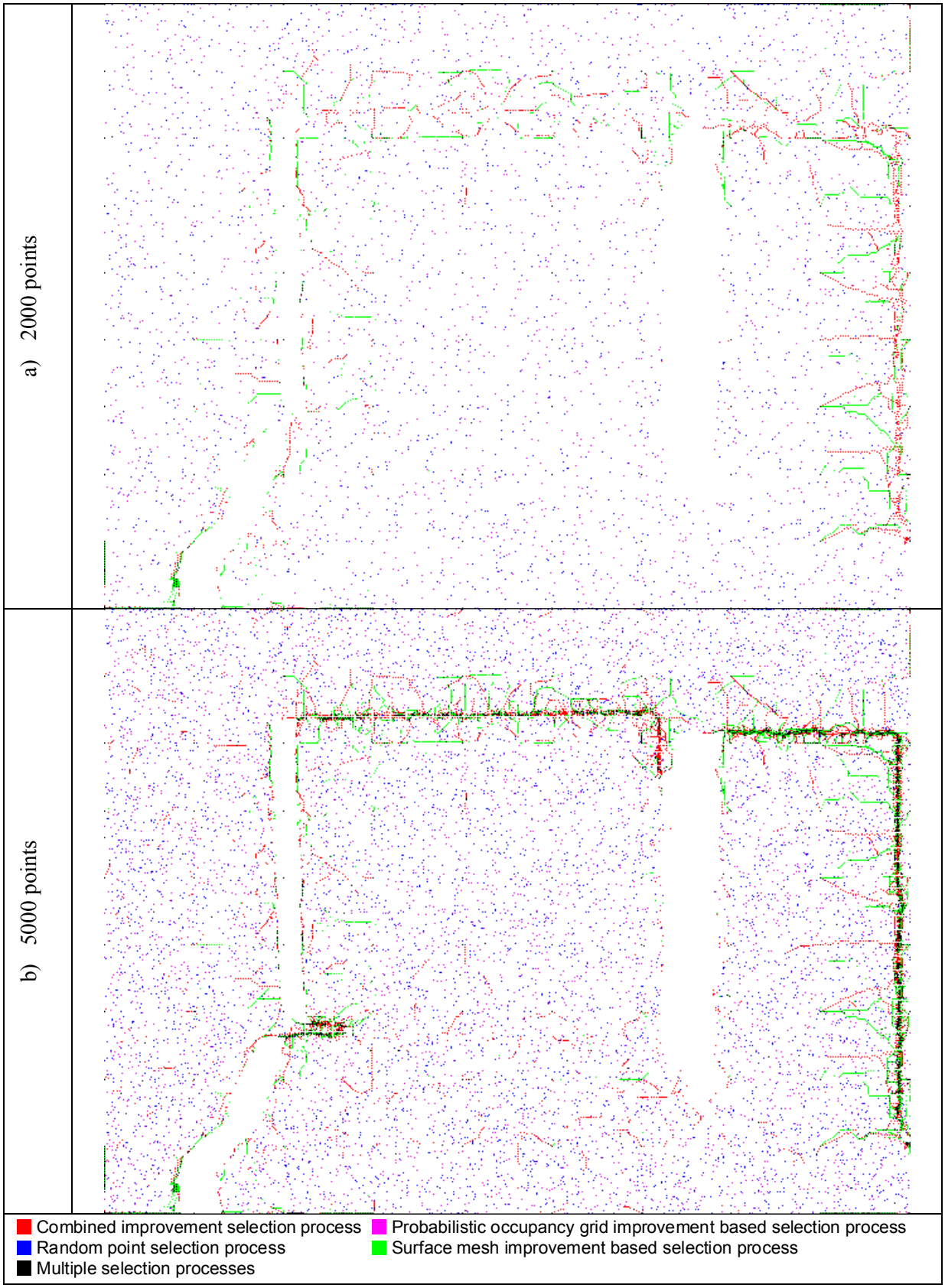


Figure D.25 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the Kinect on the computer scene (#11)

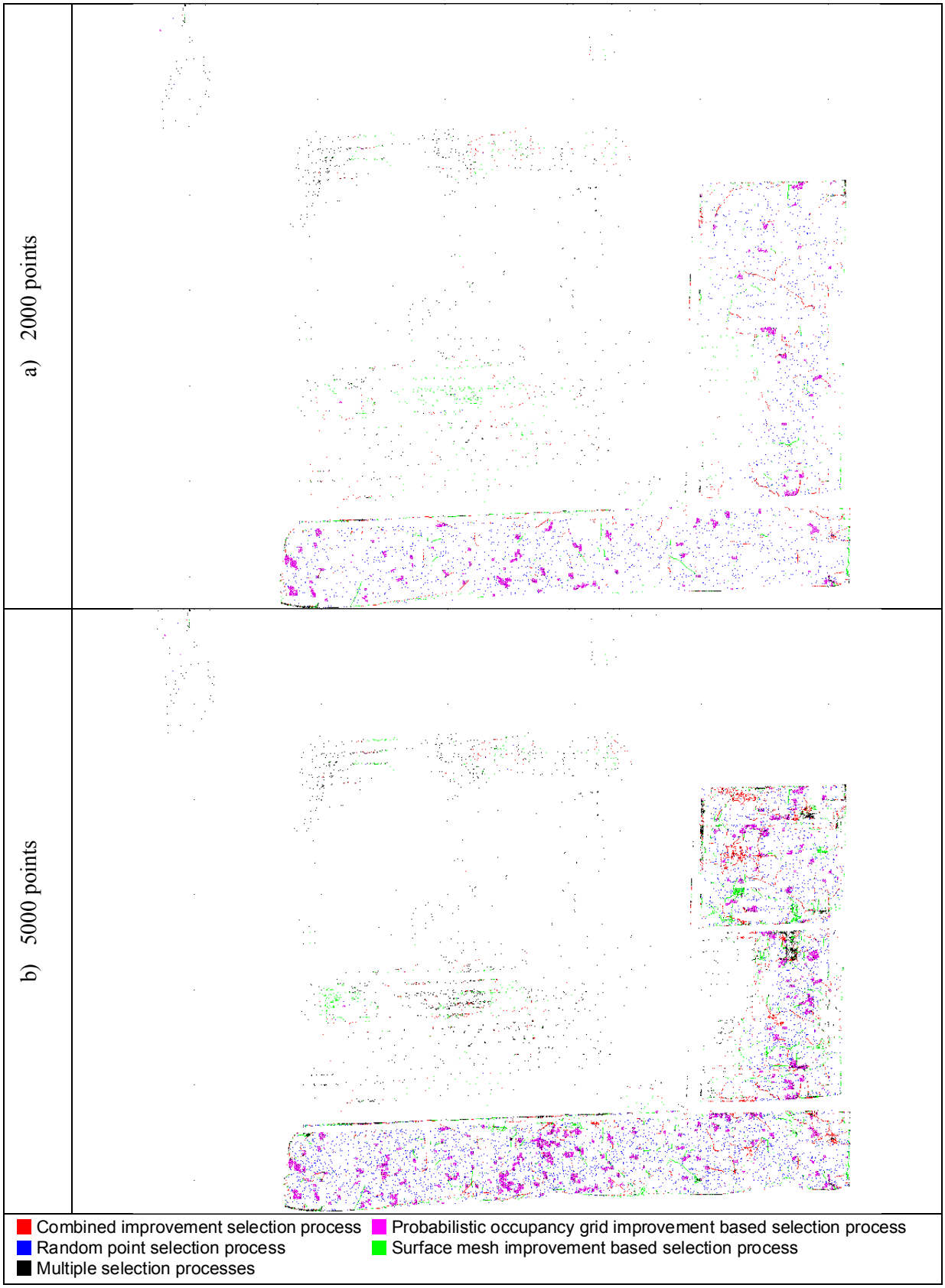


Figure D.26 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the ASLS on the computer scene (#11)

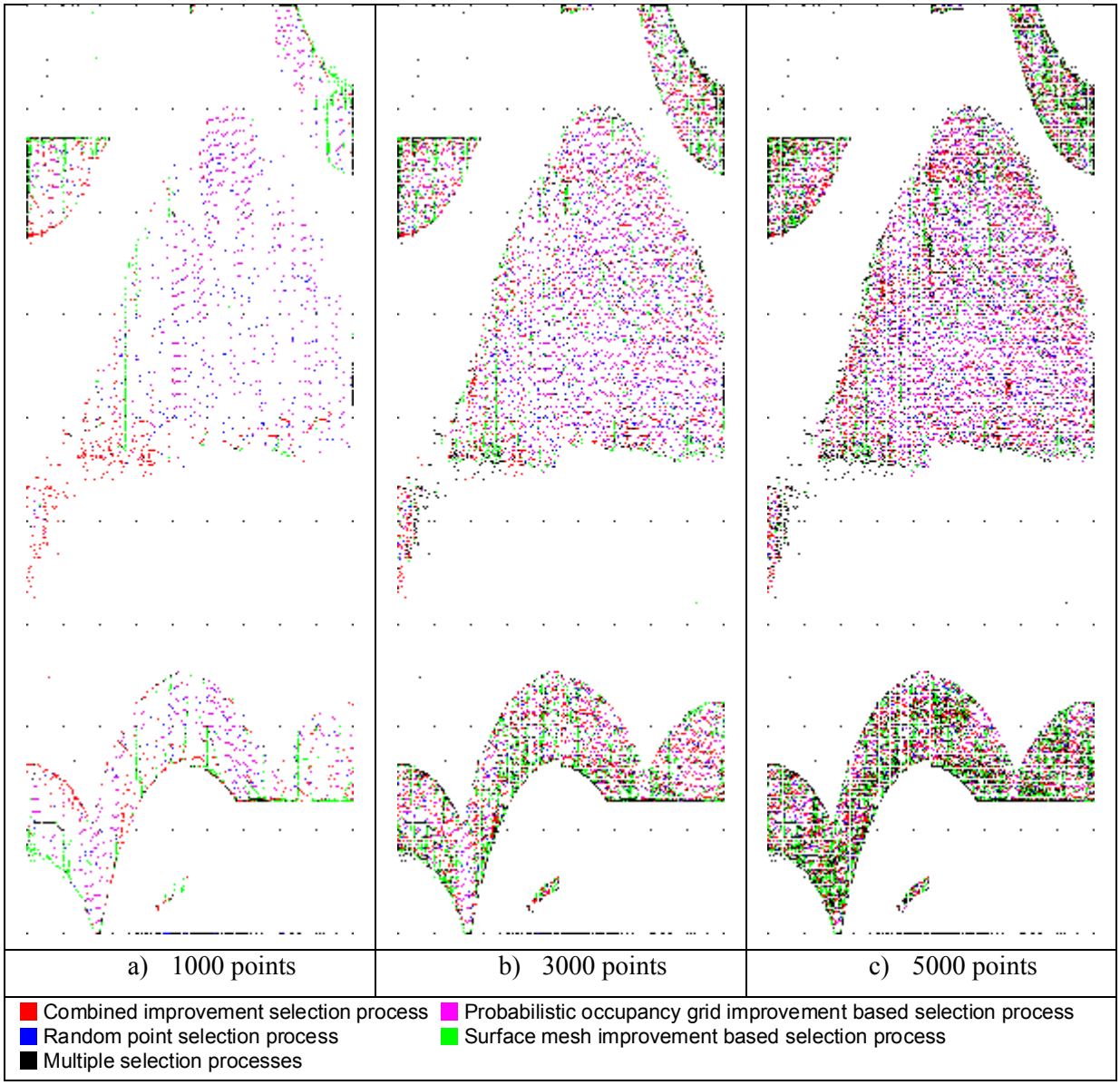


Figure D.27 - Comparative progression for a) 1000 points, b) 3000 points, and c) 5000 points selectively acquired with the Jupiter mounted on the F3T robot on the computer scene (#11)

Figure D.28 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the Jupiter mounted on the linear motorized track on the computer scene (#11)

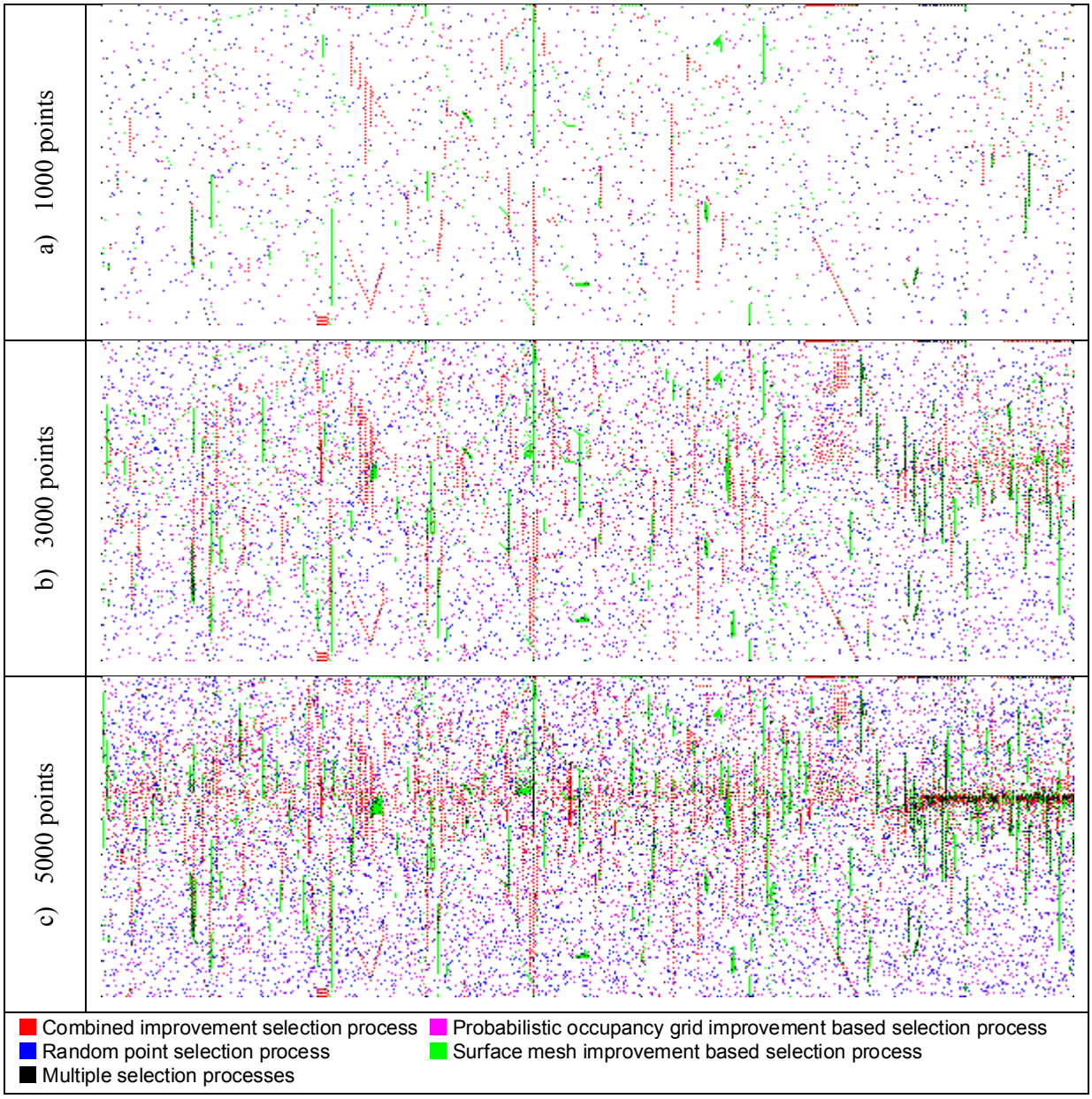


Figure D.29 - Comparative progression for a) 1000 points, b) 3000 points, and c) 5000 points selectively acquired with the SICK LMS mounted on the linear motorized track on the computer scene (#11)

D.7.2 Scene 13: Fluidly Deformed Sheet

Figure D.30, which compares the progression of the locations chosen by the four selection processes over the fluidly deformed sheet scene (Scene 13) as captured by the Kinect sensor, illustrates that the combined improvement selection process (red) performs better at covering the major features with fewer points than the surface mesh improvement based selection process (green), as it has acquired around the edge of the deformed object, as well as acquired some locations within the concave deformation, while the latter process has mainly

acquired locations along the edge only. As the number of points acquired progresses from 2000 (Figure D.30 a)) to 5000 points (Figure D.30 b)), the combined improvement selection process has better selected the various features in the scene than the surface mesh improvement based selection process. The other two selections processes, the random point selection process (blue) and the probabilistic occupancy grid improvement based selection process (magenta), have acquired locations that cover the scene uniformly, for both cases presented in Figure D.30.

The comparative progression of the point locations chosen by the selection processes for the fluidly deformed sheet as acquired by the ASLS is presented in Figure D.31. The combined improvement selection process (red) appears to have chosen points that cover the outside edges of the various regions that are identifiable in Figure D.31 a), while the surface mesh improvement based selection process (green) acquired fewer edge regions, but with clustered groups of points. After 5000 points, Figure D.31 b), these two selection processes have selected more points along the features of the regions in the scene. The random point selection process has chosen locations (blue) that uniformly cover the scene, while the probabilistic occupancy grid improvement based selection process has chosen denser clusters of points (magenta) that uniformly cover the scene. As 5000 points are added into the scene, the random point selection process also appears to have acquired clustered groups of points. These clusters must be the result of the underlying patterns projected on the scene in the ASLS acquisition scheme.

Figure D.32, which details the comparative progressive choosing of locations by the selection processes on the fluidly deformed sheet scene as acquired by the Jupiter sensor mounted on the F3T robotic arm, demonstrates that the combined improvement selection process (red) provides more coverage of the edges of features than the surface mesh improvement based selection process (green), as can be observed by the increasing quantity of red pixels versus green pixels along the edges as the locations acquired progress from 1000 points (Figure D.32 a)), to 5000 points (Figure D.32 c)). Furthermore in all three cases presented, the uniform coverage that the other two selection processes provide (the random point selection process (blue) and the probabilistic occupancy grid selection process (magenta)) is put into evidence.

The progressive captured points location maps of the scene as acquired by the Jupiter mounted on the linear motorized track are presented in Figure D.33. When 2000 points have been chosen (Figure D.33 a)), the combined improvement selection process (red) has directed the acquisition of points along the top edge of the scene, along the edge of the lower central protrusion of the scene, along the concavities present along the corners of the objects, as well as along the noisy line acquisition in the middle and the right of the scene, which provides better coverage of the features of the scene compared to the surface mesh improvement based selection process (green) that has only acquired points along the top right of the scene, the noisy scan lines along the right and middle of the scene, and some of the concavity on the bottom left of the scene. By the time 5000 points (Figure D.33 b)) have been chosen, the surface mesh improvement based selection process has acquired points along more of the features present in the scene, but much less of the features compared to the combined improvement selection process. In both cases, the random point selection process (blue) chose points that uniformly cover the scene, while the probabilistic occupancy grid improvement based selection process (magenta) has performed the same, except that points tend to be organized in 'dotted' line patterns.

Finally, the maps detailing the progressive comparison of the point locations chosen by the four selection processes on the deformed sheet scene as acquired by the SICK LMS mounted on the linear motorized track are shown in Figure D.34 for a) 1000 points, b) 3000 points, and c) 5000 points. Note that as previously discussed, the vertical axis appears to be compressed due to the lower sampling rate along the vertical axis, as well as the larger field of view, than the horizontal axis. This leads to the deformed sheet region occupying the vertical middle third of the scene. The combined improvement selection process, as evidenced by the red vertical 'dotted' line patterns, is capturing more of the vertical transitions present in the scene after 1000 points (Figure D.34 a)), compared to the surface mesh improvement selection process (green). After 3000 points (Figure D.34 b)) have been chosen, the combined improvement selection process has captured much of the top of the object, as well as some of the bottom, while the surface mesh improvement process has captured fewer segments in these regions. Finally, after 5000 points (Figure D.34 c)), the combined improvement selection process has mostly captured the top of the deformed object, where the deepest transition of the scene lies, while capturing locations more sparsely about

the middle of the scene. The probabilistic occupancy grid improvement based selection process (magenta) and the random point selection process (blue), in all three cases, have chosen to acquire points that uniformly cover the scene.

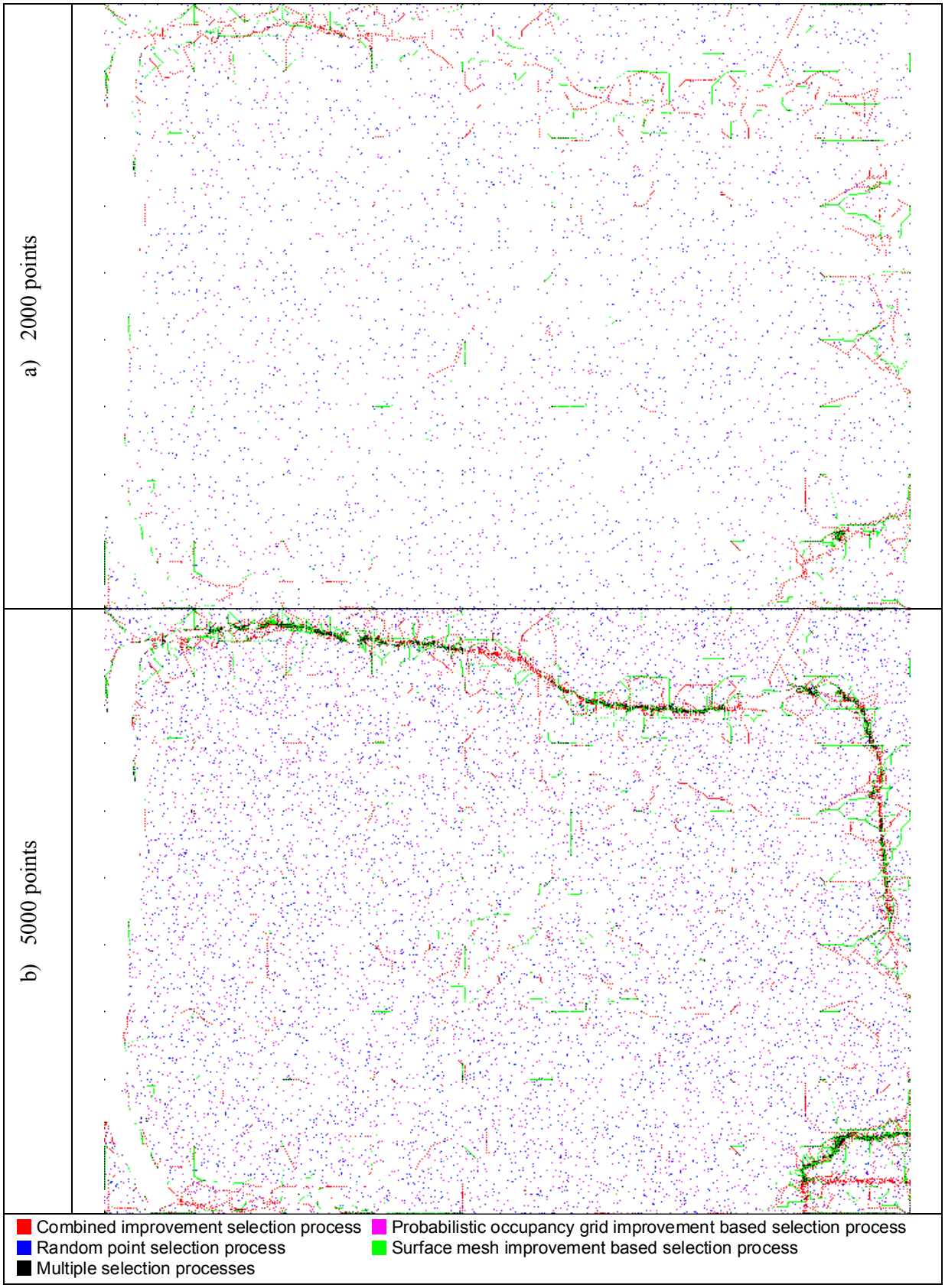


Figure D.30 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the Kinect on the deformed sheet scene (#13)

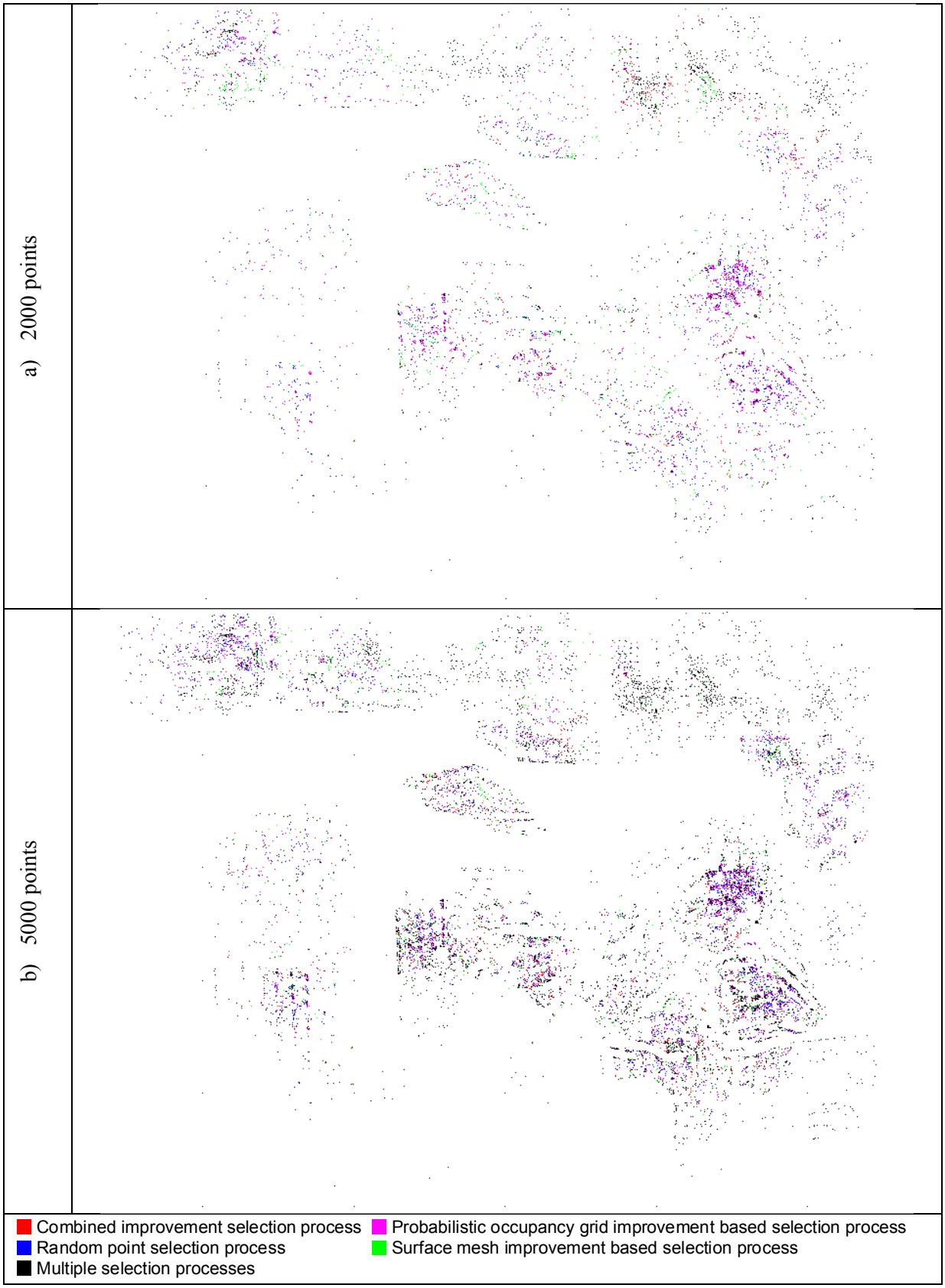


Figure D.31 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the ASLS on the deformed sheet scene (#13)

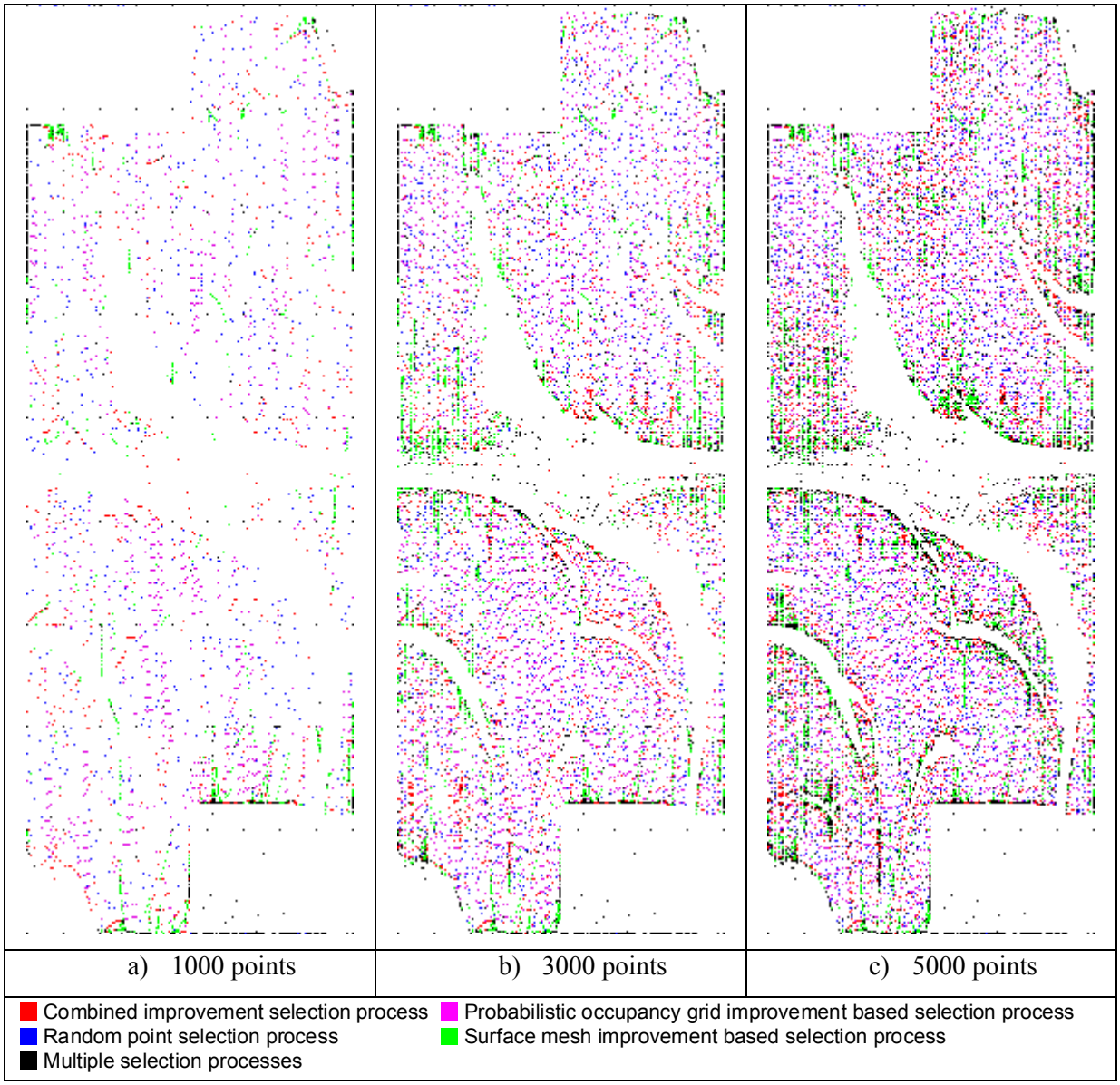


Figure D.32 - Comparative progression for a) 1000 points, b) 3000 points, and c) 5000 points selectively acquired with the Jupiter mounted on the F3T robot on the deformed sheet scene (#13)

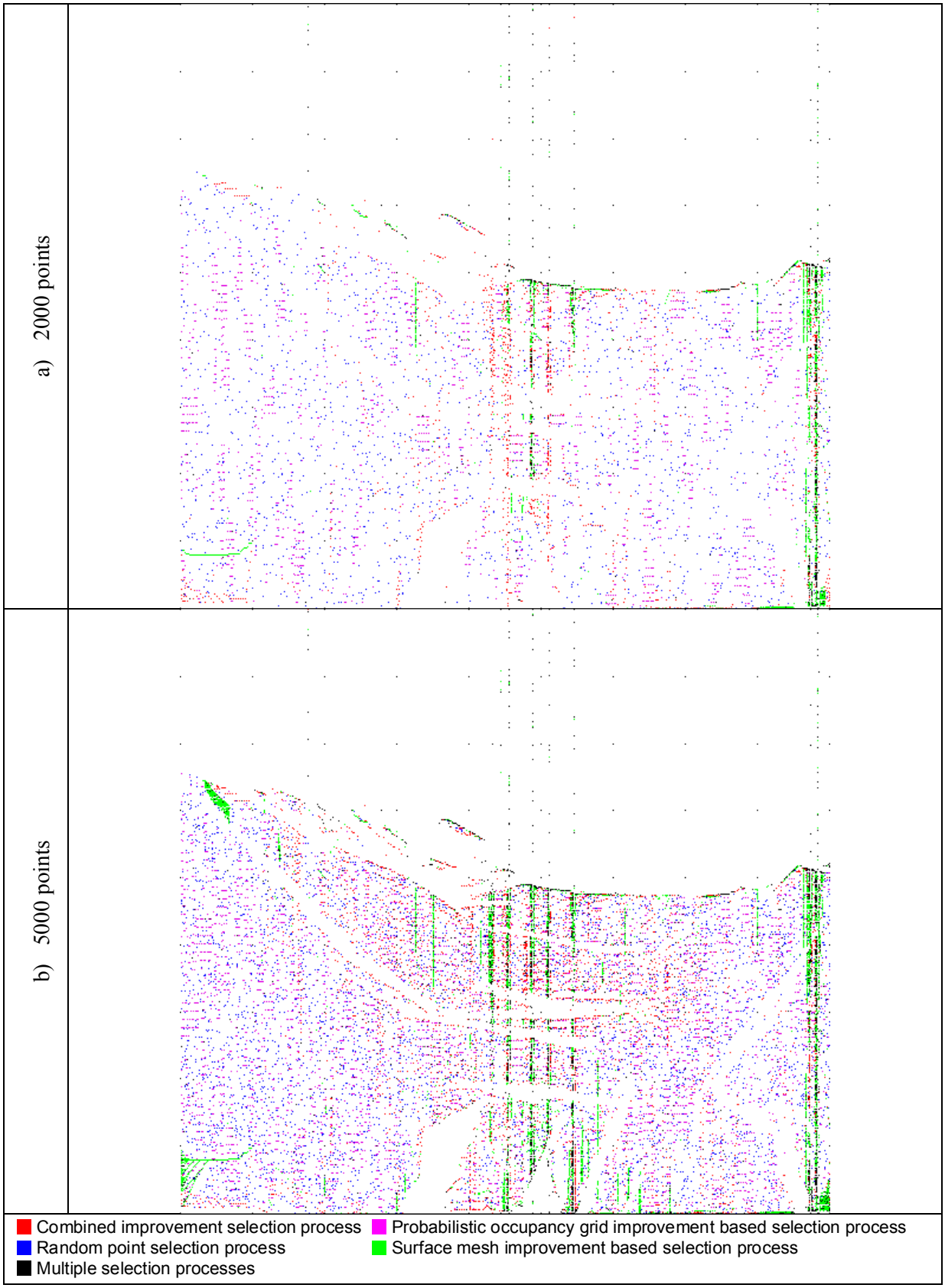


Figure D.33 - Comparative progression for a) 2000 points, and b) 5000 points selectively acquired with the Jupiter mounted on the linear motorized track on the deformed sheet scene (#13)

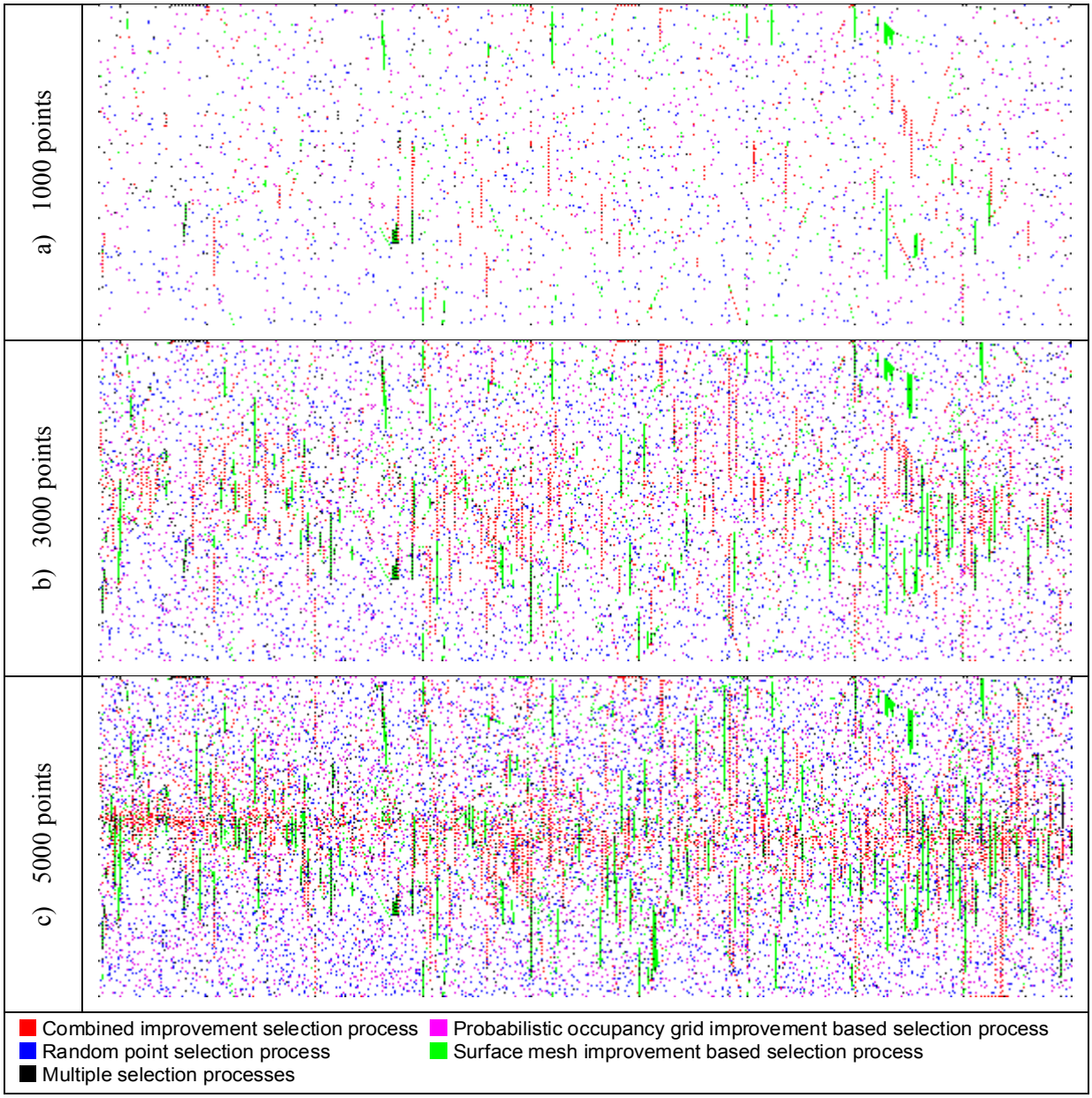


Figure D.34 - Comparative progression for a) 1000 points, b) 3000 points, and c) 5000 points selectively acquired with the SICK LMS mounted on the linear motorized track on the deformed sheet scene (#13)