Aberystwyth University

Doctoral Thesis

Environment perception in the context of 3D terrestrial laser scanning

Author:
Marek Ososinski

Supervisor:
Dr. Frédéric Labrosse

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Field and Applied Robotics Team
Department of Computer Science

March 2016

Declaration of Authorship

DECLARATION

This work has not previously been accepted in substance for any degree and is not being currently submitted in candidature for any degree

Signed: _____

Date: _____

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references, A bibliography is appended.

Signed: _____

Date: _____

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and the title and summary to be made available to the outside organisations.

Signed: _____

Date: _____

ABERYSTWYTH UNIVERSITY

Abstract

Institute of Maths, Physics and Computer Science
Department of Computer Science

Doctor of Philosophy

Environment perception in the context of 3D terrestrial laser scanning

by Marek Ososinski

Terrestrial laser scanning has become a popular way of digitising buildings and complex environments. Laser scanning was adopted as the means of capturing 3D data in many fields, including architecture, engineering and environmental survey. It was only a matter of time for the Heritage sector to start using the technology. This thesis describes the scientific contributions from the collaboration project that explored the viability of automating the laser data acquisition process. The project concentrated on the reduction of the skill set required by the operator of the laser scanner as well as the improvement of the usability of large datasets. The contributions involved the development of a new data representation method, a new visibility estimation metric and an improved volumetric decimation algorithm.

Acknowledgements

First I would like to thank my supervisors Frédéric Labrosse and Mark Neal for the encouragement, support and countless hours spent discussing the ideas and their implementation.

Next my thanks go to Susan Fielding and Louise Barker for somehow managing to cope with me during the meetings and in the field.

To everyone in the Computer Science Department at Aberystwyth University. Especially to Alexander Spence for constant support throughout the years.

There are plenty of others without whom I would not be able to finish this PhD: Pete, Shisheng, Alex and Chris. You kept me just insane enough to make it through.

Last but not least I would like to thank my parents for making it possible in the first place and my grandparents for introducing me to science.

This project has been supported by the European Social Fund through the Welsh Government as a part of the KESS program.

Contents

List of Figures

List of Tables

For Janusz Warchol, may his legacy live forever.

Chapter 1

Introduction

1.1 Background and motivations

Terrestrial laser scanning has become a popular way of digitising buildings and complex environments. With the variety of hardware models available, laser scanners can provide anything from sub-millimetre resolution, through to several kilometre range, to colour data. Laser scanning was adopted as the means of capturing 3D data in many fields, including architecture, engineering and environmental survey. It was only a matter of time for the Heritage sector to start using the technology.

The uses of laser scanning in the Heritage sector include preservation, restoration, survey and monitoring. Preservation work relies on digitising the location *as is*, to capturing the state as well as feel of the building. Such work is often done pre-emptively, to present the building in its innate state, showing how it would be used during every day life. Sometimes preservation work is done before demolition of the building to preserve by record what is left from the original. Digitisations are used for the restoration of locations, including demolition and creation of facilities as well as renovation of architectural features and plasterwork. Scans are sometimes used for the recreation of hard to copy features such as mouldings. The most common practice, however is a visualisation of *before and after* comparison, including the original state of the building, planned work and the final result of the renovation. Survey work uses the digitisation to evaluate the current state of the building. Laser scans are often better at showing the scale of physical degradation than photos. Monitoring of the site can be achieved by repeated survey that can lead to the understanding of the degradation of the building.

Terrestrial laser scanning is not always the answer, it comes with many drawbacks including the cost of hardware, time necessary to capture and process data, vast amounts

of data captured and requirement for skilled workers on site. Additionally, due to the broad spectrum of applications, such as covering vast areas many kilometers from the origin, capturing architectural features at sub-milimetre resolution, gathering elevation data from aerial viewpoint, detecting the depth of vegetation in a given area, etc.; laser scanners are built with specific tasks in mind. This leads to the lack of standards for data representation and processing and inhibits any attempts at automating the process.

Photogrammetry is a cheaper and more intuitive alternative to laser scanning, that has been proven to work in field conditions [1]. Therefore it could be an obvious choice for the digital reconstruction. However, the main drawbacks of photogrammetry are unreliability in low lighting conditions, high dependence on atmospheric conditions, scale and resolution issues and self occlusion handling problem. Photographs taken in low lighting condition tend to have low colour variance and are problematic during image matching, which leads to unusable samples. Mist and rain make most photographs unusable and the sun or any bright light can cause saturation of photographs. The most extreme case is a dark room with a single window letting the sun in. In such a room the contrast in the luma greatly inhibits image matching. Photogrammetry works well on artefacts, buildings and other standalone or similar scale structures, however attempting to extract a gargoyle from a picture of a cathedral might prove difficult. Indeed, assuming that high resolution pictures of a gargoyle were explicitly taken using a telephoto lens, both the lens geometry and the atmospheric effects such as dust, heat waves and humidity would alter the appearance of the gargoyle by distorting it and softening the image. This would make it hard to match the gargoyle with the low resolution image of the cathedral. High resolution images require vast amounts of processing power to match and processing time increases exponentially as the dataset grows in size. Last but not least, photogrammetry does not work well in concave environments such as rooms and caves due to the change in perspective. In such environments narrow field of view lenses are often struggling to capture enough key points to match the consecutive images together. Use of a wide angle introduces heavy distortions. Additionally in low light conditions additional static lighting is required. Photogrammetry struggles with concave features of objects, requiring additional samples from different viewpoints to mitigate the issue.

All in all laser scanning does not suffer from the aforementioned problems as much as photogrammetry does, therefore it provides more consistent results. It does introduce a new issue: highly reflective or light absorbant materials distort the laser causing the resulting values to be incorrect, introducing noise. Even though laser scanning was chosen as a primary means of reconstruction the two are not mutually exclusive and can be used in conjunction.

1.1.1 Laser scanning

The process of laser scanning can be divided into four phases: reconnaissance, position detection, scanning and registration. During reconnaissance an operator is scouting the area for the possible scanner placement positions. After the environment has been explored the operator prepares a scan strategy, selecting the scanner positions while considering major occlusions, accessibility and allowed time. This stage often takes upwards of an hour depending on the environment and directly leads to the detection of the positions from which the separate laser scans will be taken. After the planning stage, scanning starts. Each scan takes from 15 to 20min depending on the selected resolution[1]. The time taken to transport the scanner between scans is directly linked to the accessibility of the next scanning position. The last stage involves registration of the scans, where common planes and points are designated by the operator and scans transformed to match a common coordinate system. In the case of lack of features in the scene artificial targets can be placed to ease the detection of corresponding points. Registration is a process of registering all the scans to a common origin, by applying a transformation to all the points within the scene. Combining all the scans to create a complete scene is performed semi-automatically by selecting corresponding points in multiple scans and aligning the scans based on the planes created by those points. Alternatively, if the (GPS) coordinates of enough corresponding points are known the process can be fully automated.

The main trade-offs of laser scanning include time taken on location, number of scans and processing time. A well planned scan can save a considerable amount of processing time, each additional scan requires at least 3 common reference points with every overlapping scan to ensure the correct registration. The geometrical complexity of the scene will directly influence the number of scans taken, a more complete model will require larger amouts of scans directly influencig the time spent on location. Each additional scan has a potential to introduce more noise into the registration, the small errors caused by the accuracy of the instrument are multiplied by the inaccuracy of the transformation. This is however better than not having enough scans to provide sufficient data overlap to correctly register the scans. The main way of improving the efficiency is to limit the number of scans while maximising their coverage. Finding the exact amount of scans can prove difficult.

[1] An average scan time taken by Leica HDS6200 scanner to acquire a scan at high resolution.

1.1.2 Environments

Terrestrial laser scanning is used to capture models from a range of different environments. This includes natural formations such as forests, caves and riverbanks, as well as altered environments such as fields, quarries and man made structures such as roads, buildings and castles. Based on the general shape we classify the scenes into outdoor (convex) and indoor (concave) environments, where an outdoor environment is one that has open bounds or a horizon and an indoor environment is one that is, at least partially, enclosed. This division is caused by the vastly different approaches of digitising convex and concave shapes. The next division is between artificial structures, which often have defined planes and corners, and natural environments that are more organic. Conventionally man-made structures have multiple points that could be used as reference points and rarely require the use of external targets. On the contrary, natural environments rarely have distinct features that could be used as reference and require the use of additional registration targets. It is not uncommon that the environment being digitised contains a mixture of natural and artificial features, often containing structures that should be digitised both on the outside and inside. Many environments have additional restrictions such as accessibility issues or become hazardous during harsh weather condition. Ideally, all environments could be approached in a simple, methodical way, that would help the operator scan the environment. Unfortunately, often, the more interesting the location is the harder it is to scan it!

The most commonly scanned locations in our case studies are man made structures, ancient and relatively modern alike. In addition to aforementioned factors such structures have a physical state and proprietary restrictions, often limiting the type of equipment that can be used. Some structures can be in a relatively poor state, ranging anywhere from rotten floors to danger of collapsing. In such case it is essential to reduce the time an operator has to spend on location.

1.2 Problem summary

We were approached by the Royal Commission on the Ancient and Historical Monuments of Wales to develop a system capable of improving the efficiency of laser scanning by the introduction of automation into the process. Environmental surveying involves multiple media, laser scanning being just a small part of the whole process. It would be beneficial for a novice to perform a good quality laser scan without the need for extensive training. The main objective of the collaboration is to allow an untrained operator to scan any environment in a limited time frame whilst ensuring the completeness of the coverage.

The time allocated to data capture is counted in half-day increments, most common being a day long survey. The scanning process requires an expert operator despite being very repetitive with short periods of interaction intertwined with long periods of inactivity. An automated system would allow novice operators to slowly build their expertise while not devoting all their attention to laser scanning, eliminating the need for an expert in the field.

Laser scanning is a simple process in which the operator is placing a laser scanner in a position where it would have the best line of sight with the environment all around that position, then moving the scanner to another location and repeating the process. It is very similar to taking omnidirectional pictures. As the process is comparable to taking pictures, it seems natural that a human would be good at positioning a laser scanner in correct positions. This is true for convex shapes, such as trees, buildings, old ruins, etc. However, making a comprehensive record of a concave shape, such as a cave, a room or the inside of a box is not as straight forward. This is caused by the way human beings think about enclosed spaces: indoors the perspective changes, wider views becomes more distorted and parallel lines seem non-parallel. To compensate, human beings tend to think about enclosed spaces using topological maps and planar projections [2], whether it is a floor plan of a room, or a cross section of a cave. Moreover human beings are not good at judging the scale of an environment as they often lack points of reference [3]. Combining all that, humans are fairly bad at estimating occlusions within enclosed spaces. An added difficulty is the omnidirectional nature of the laser scans; we tend to think of pictures as rectangular viewpoints, not spherical projections. An arched ceiling or centuries old beam are equally as important as a mural on the wall or a statue in the corner. Unfamiliar perspective and inherent inability to consider multiple occlusions make laser scanning difficult.

The seemingly daunting idea of automating the process of laser scanning can be divided into smaller, more manageable chunks. We propose that the process involves crude mapping of the environment, spatial analysis and detection of the scan positions, physical placement of the scanner and scanning. Environment mapping has been a hot topic for many years. Both vision based techniques and range based techniques are being adapted to cope with full 3D representations of the environments. Due to time constraints on the project no new developments were made for the environment reconstruction, which was only briefly explored, and ultimately only critically discussed in Chapter 7. Spatial analysis, however, is a relatively neglected area due to the difficulties of working with 3D data. The detection of scanner position is a task requiring the development of spatial reasoning techniques, and potential improvements to 3D data understanding and representation. This stage became the primary target of the presented research.

Physical placement of the scanner is more of an engineering challenge and was left out from the process due to various factors, described in Chapter 7.

To begin the consideration of spatial reasoning some representation of an environment is required. We considered using topological maps and 2D metric maps, none of them were able to retain enough information about the nature of the environment. A 3D representation was the chosen solution.

1.2.1 Improving 3D data representation to enable spatial reasoning

One of the primary challenges is the processing of 3D data. The most basic representation is an unordered list of points. Such representation, albeit efficient, is very hard to use in the context of spatial reasoning. A simple list allows processing of each point individually, but not in relationship with other points. To represent spatial relationships between points the dataset can either be converted to a mesh, therefore providing information about nearest neighbours of the point and creating polygons, or converted into a volumetric representation by delimiting the points using an equidistant grid of cubes. The mesh based representation provides faces and ease of computation of angle of incidence for visibility estimation, however it lacks spatial relations between non-neighbouring points. This makes occlusion detection and ray casting unnecessarily expensive. Volumetric representation is very good at presenting spatial relations between points, but lacks information about face direction. The volumetric representation shows only the presence of a point within an arbitrarily axis aligned space. The surfaces that are not axis aligned become jagged and as such angles of incidence are hard to acquire.

1.2.2 High resolution multi viewpoint visibility estimation using reduced resolution datasets

The choice of data representation is supposed to aid the detection of scanner positions within an environment. To estimate the visibility coverage of each viewpoint, a set of potential positions of a scanner during the data acquisition process are placed within a virtual environment. Each viewpoint is evaluated using our visibility measure and the existing set of multiple viewpoints is then reduced to a more desirable solution if needed. The visibility estimation is based on the distance and angle of incidence between a viewpoint and a face of a voxel. A novel volumetric representation was used to aid the computation of visibility, one that allows for efficient ray casting, while providing sufficient surface orientation information. The low resolution datasets prove sufficient to estimate a viable solution for a multi viewpoint system. To support the viewpoint

viability estimation a new method for visibility estimation was developed and will be further described in chapter 5.

1.3 Hypothesis and research question

The hypothesis for this thesis is:

> "It is possible to eliminate the reliance on expert knowledge in terrestrial laser scanning by partially automating the process."

The research question for this thesis is :

> "Is a low resolution representation of a real-world 3D environment sufficient to approximate the completeness of a multi viewpoint visibility estimation of that environment?"

The hypothesis is exploring the possibility of capturing, and replicating the expert knowledge in order to allow any operator to perform quality, cost effective scans.

1.4 Objectives

1. Automation of laser scanner position detection – Although full automation is not yet viable, see Chapter 7, parts of it requiring expert input can be semi-autonomous. The desired outcome is a shift of the spatial analysis of the environment and position placement detection away from the operator, to improve the reliability and quality of the scans performed by the less experienced users.

2. Optimisation of the number of required scans – Number of scans is the main factor in the time taken on location and directly influences the workload during scan registration. We aim to reduce the number of scans while ensuring scan completeness.

3. Analysis of the laser scanning process – The process of laser scanning includes planing, scanning on location, and data registration and processing. We want to

explore the different approaches to laser scanning and replace some of them with an automated system.

4. Improvement of the usability of resulting large point clouds – Point clouds generated by terrestrial laser scanning are often beyond the processing capabilities of consumer hardware, to enable the interaction with the datasets we propose a decimation method.

Those improvements should allow a wider range of people to work with point cloud data including laser data acquisition which should lead to further improvements in the field.

1.5 Thesis outline

- Chapter 2 provides a further analysis of the laser scanning process including the description of sample locations
- Chapter 3 describes the state of the art of representing environments in the context of spatial reasoning and visibility coverage estimation.
- Chapter 4 provides an analysis of spatial perception and reasoning including the analysis of environment perception, highlights the importance of correct data representation and understanding, and describes a new volumetric way of representing 3D data that aids spatial reasoning.
- Chapter 5 presents the new multi-viewpoint visibility coverage estimation. The proposed visibility estimation is capable of approximating the visibility coverage at a higher resolution, providing a slight underestimate of the visibility as the resolution decreases. The viewpoint position detection is providing a set of viewpoints within the given environment that is of comparable quality to a set chosen by a human operator.
- Chapter 6 describes a fixed point density volumetric point cloud decimation algorithm that allows a point by point density reduction in a volumetric manner, without the need for creating an octree based representation.
- Chapter 7 is a discussion about the presented methods involving their requirements, benefits and drawbacks, as well as evaluating the viability of using them in the real world. It also concludes the thesis, reiterating the main objectives and achieved results with additional comments.

  Experimental results are given at the end of each relevant chapter.

1.6 Publications and attended Conferences

The following conferences were attended during the project to present the relevant findings:

- International Conference on Computer Vision Theory and Applications (VISAPP) 2014, Lisbon, Portugal
- RIVIC Graduate School 2013, Bangor, UK
- Digital Past 2013: New technologies in heritage, interpretation and outreach, Monmouth, UK
- Robotics innovation for Cultural Heritage (RICH) 2012, Venice, Italy
- Digital Past 2012: Digital Technologies and Heritage, Lllandrindod Wells, UK

One conference publication was made during the project:

Marek Ososinski, Frédéric Labrosse. *Multi-viewpoint Visibility Coverage Estimation for 3D Environment Perception — Volumetric Representation as a Gateway to High Resolution Data.* VISAPP (2) 2014: 462-469

This publication covers the visibility measure in Chapter 5.

An additional two journal papers are currently under preparation.

Chapter 2

Analysis

2.1 Project description

The Royal Commission on the ancient and historical monuments of Wales is the investigation body and national archive for the historic environment of Wales. Its purpose is the archiving and promotion of the archaeological, built and maritime heritage of Wales. The Royal Commission is constantly exploring new technologies allowing the preservation of the archival record in increased detail. The collaboration between the University and the Royal Commission involved the exploration of laser scanning as a mean to digitally preserve a multitude of locations across Wales. With limited resources and high workload a cost effective solution for acquisition of laser scans was desired. This collaboration resulted in a project attempting to automate, or at least optimise, the process of laser scanning. Our involvement included collaborative surveys of a range of locations resulting in fully registered scans, bi-directional knowledge transfer allowing to improve the skills of the involved parties and, at least partial, automation of the scanning process.

2.1.1 Four phases of laser scanning

Laser scanning can be divided into four phases: reconnaissance, position detection, scanning and registration, see Figure 2.1.

The *reconnaissance* stage starts with planning which location to scan, deciding on the desired areas of interest and allocation of sufficient time on location to complete the survey. This phase continues on location, where the environment is visually inspected.

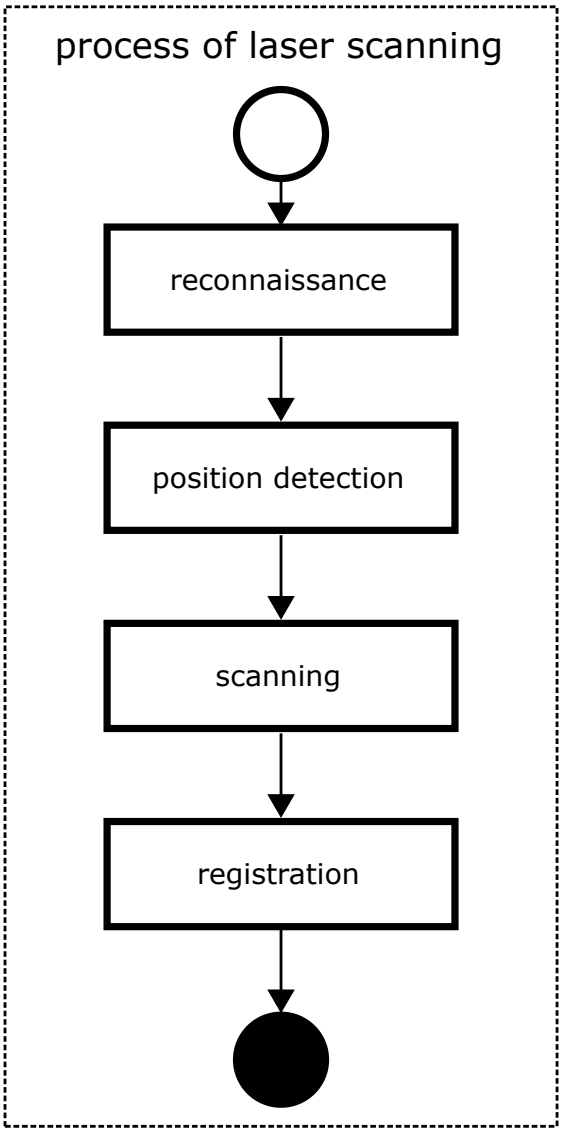


FIGURE 2.1: Four phases of laser scanning

Next, *position detection* takes place, where the operator creates a mental or written plan of the scan, deciding upon the locations of the scans.

The third phase, *scanning*, involves the positioning of the scanner to the previously chosen locations and initiation of the scan.

The final phase, *registration*, happens off-site, where the operator visually inspects the generated point clouds, determines corresponding points within pairs of scans and registers them to a common origin.

2.1.2 Survey work

A major part of the project was laser scanning of various locations. They range from a muddy river bank, through to an ancient burial site, to churches and an industrial complex. In this section the most unique of the locations that have been laser scanned are described. The laser scans were performed to digitise the various locations due to their poor state, chance of collapse or as a record before major renovation work.

Ynyslas

The wreck in Ynyslas, see Figure 2.2, is lodged inside a riverbank. The terrain surrounding the wreck is fully covered in mud, limiting movement. The location is time critical as it is flooded during high tide. The object of interest is the wreck itself.

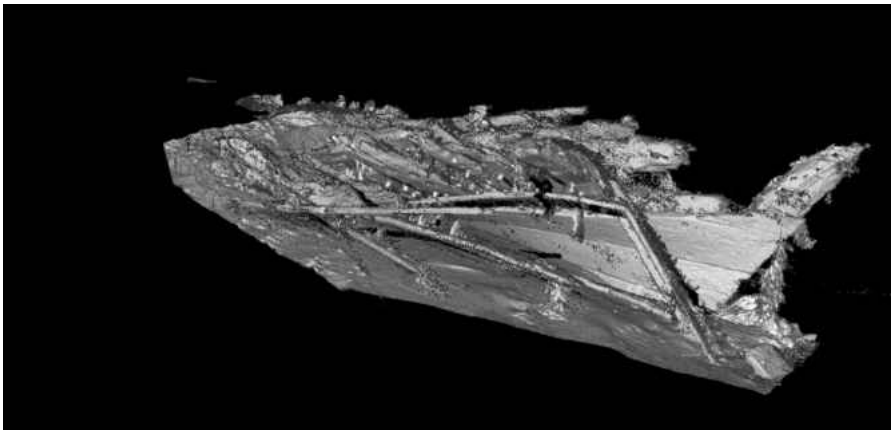


FIGURE 2.2: Wooden barge in Ynyslas

Bridge

A medieval bridge over a small river that floods from time to time, see Figures 2.3 and 2.4. The location has very limited accessibility, limiting the potential scanning positions. Those are further obstructed by trees and fences. The object of interest was the bridge itself, and especially a part that has been damaged by the flood.

Tomb

An ancient tomb consisting of several chambers lined with stone slabs and covered in dirt, see Figure 2.5. Part of the tomb was uncovered by an unfortunate attempt to extract some stones for construction purposes. The location contained several stone slabs and earthworks.

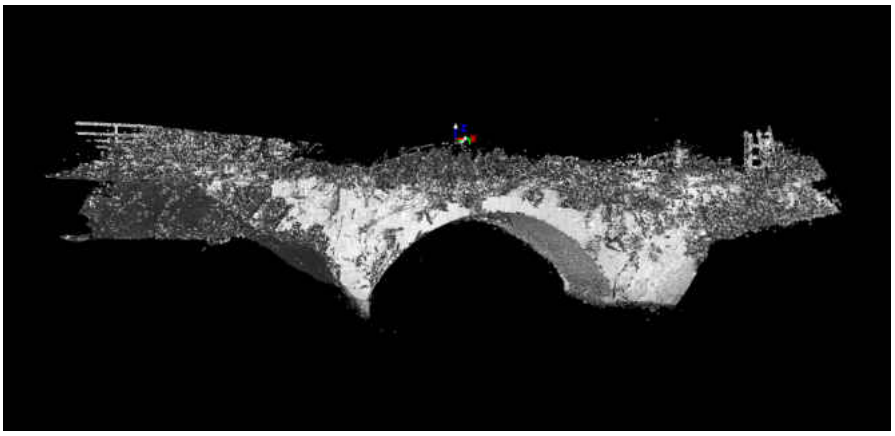


FIGURE 2.3: Bridge, south west side

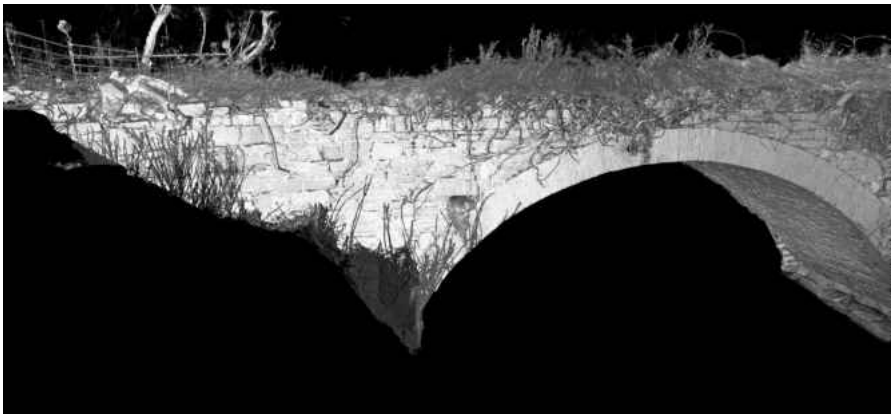


FIGURE 2.4: Bridge, north east side

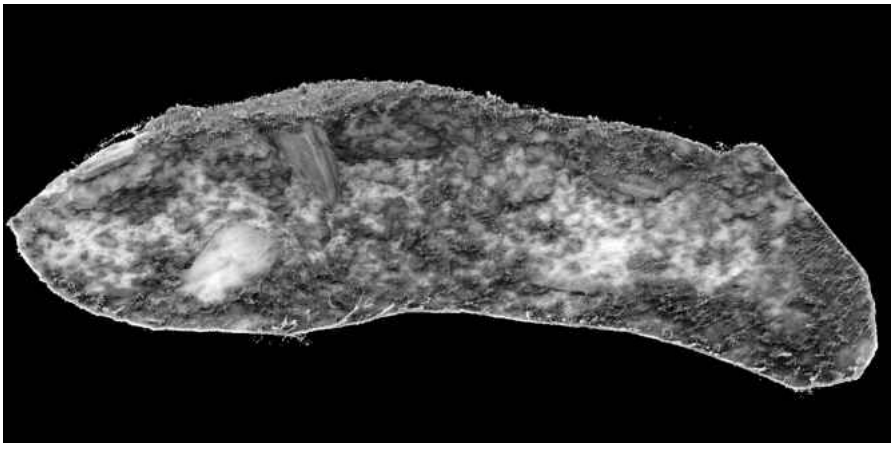


FIGURE 2.5: Ancient tomb

Grosmont

A church in Grosmont, see Figure 2.6, containing pillars, arches and a wooden beam roof in the main chamber. The point of interest was the historical roof.

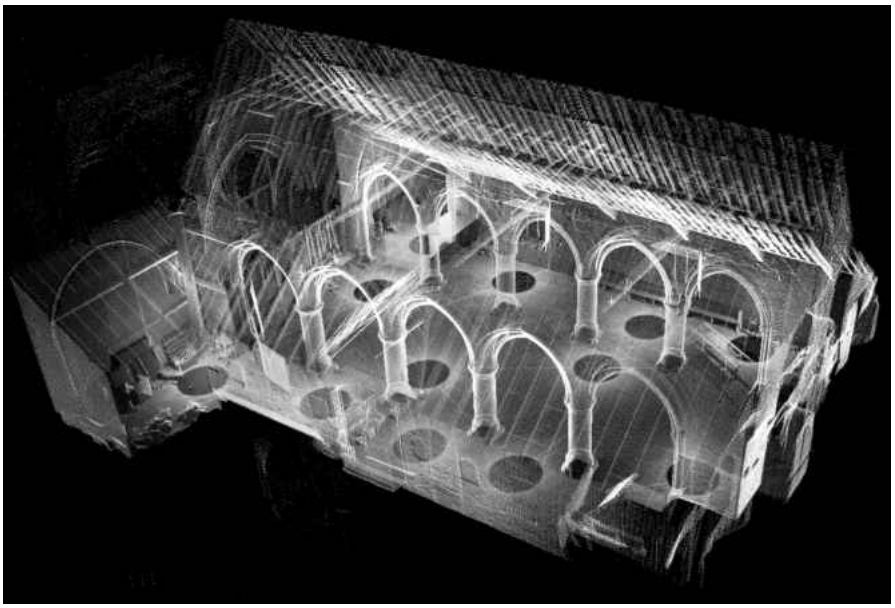


FIGURE 2.6: Grosmont church

Denbigh

A church in Denbigh that consists of two main chambers, each built in a different period, columns and arches, as well as a vast amount of furnishings and artefacts, see Figure 2.7 and 2.8.

Bethania

A large chapel consisting of a main chamber and a balcony. The main aim was to capture the remaining detail of the mouldings and complex wooden altar, see Figures 2.9 and 2.10

Brymbo Ironworks

A vast area of ironworks, consisting of multiple buildings at different elevation levels with an old blast furnace, Figure 2.11, later converted to a sand hopper, a foundry with a partially collapsed roof, Figure 2.12, steelworks building, workshop and several old housing building. The main target was a complete scan of a location and detailed scan fo the debris from the collapsed roof, see Figure 2.13.

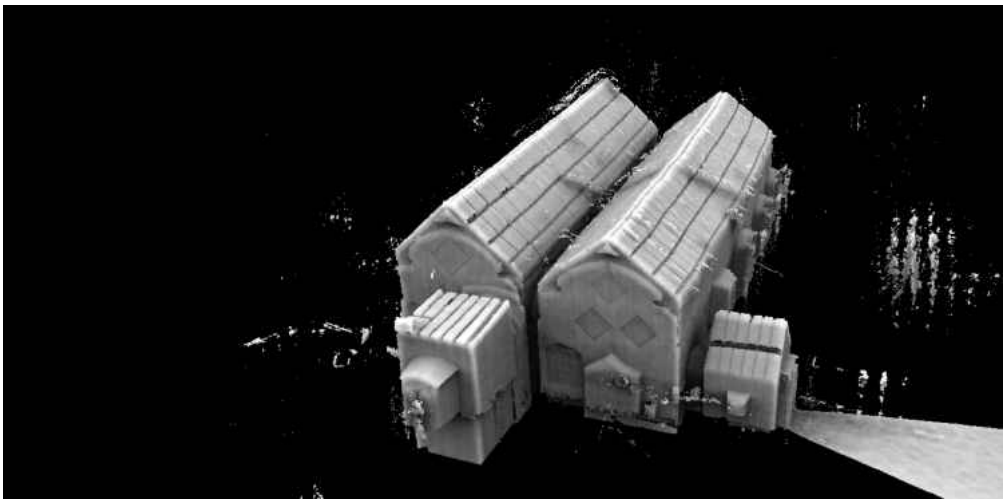


FIGURE 2.7: Denbigh church overview

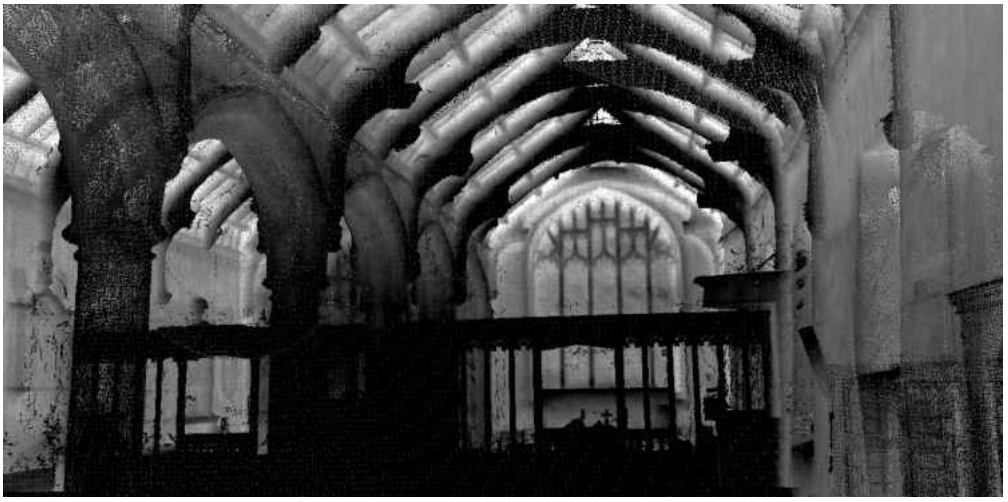


FIGURE 2.8: Denbigh church inside

2.1.3 Public engagement

Parts of the project involved community days, where people from the local area would come to explore the latest discoveries about their places of historical importance as well as learn about the latest survey equipment. Those play a crucial role in the acceptance of the surveying process. By presenting the benefits of, and exposure to the new technologies people build up trust. During the two community days we have encountered dozens of locals. Very few of the visitors had prior contact with laser scanning and were surprised by the detail of laser scans. Most compared the results to high resolution photographs, and were intrigued by the placement of the scanner on location as it was counter intuitive when compared with a placement of a camera.

During the time spent on location many historical groups and passers-by were intrigued by what a laser scanner was doing. Similarly to the community days the participants were surprised by the detail captured by the laser scanner.

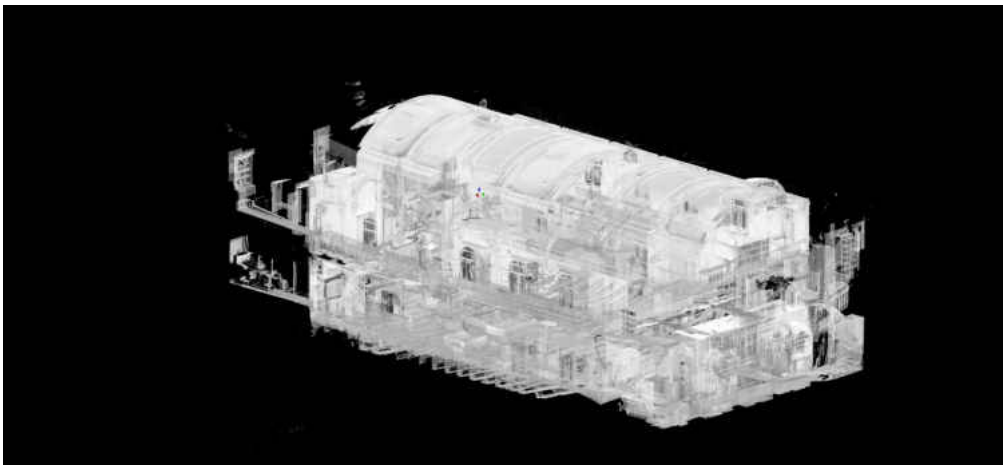


FIGURE 2.9: Bethania chapel overview

FIGURE 2.10: Bethania chapel inside

In general people without prior experience with the laser scanner approach the device the same way they would a wide lens camera and tend to try and frame a shot. The concept of a panoramic image is fully understood, but the placement of the device remains counter intuitive.

2.2 Human expertise and automation

Attempting to replicate a task performed by a human is a common problem in automation [4]. Physical actions, such as moving objects, can be tedious to automate as they require a large amount of instructions and are limited by the hardware used. However, they are well defined and straight forward to define. Tasks that conventionally require reasoning are a lot more complex and often require an engineered solution based on the knowledge of a human being. Humans often use sets of directives to perform specific tasks. Those directives are high level descriptions of actions or required evaluations. For

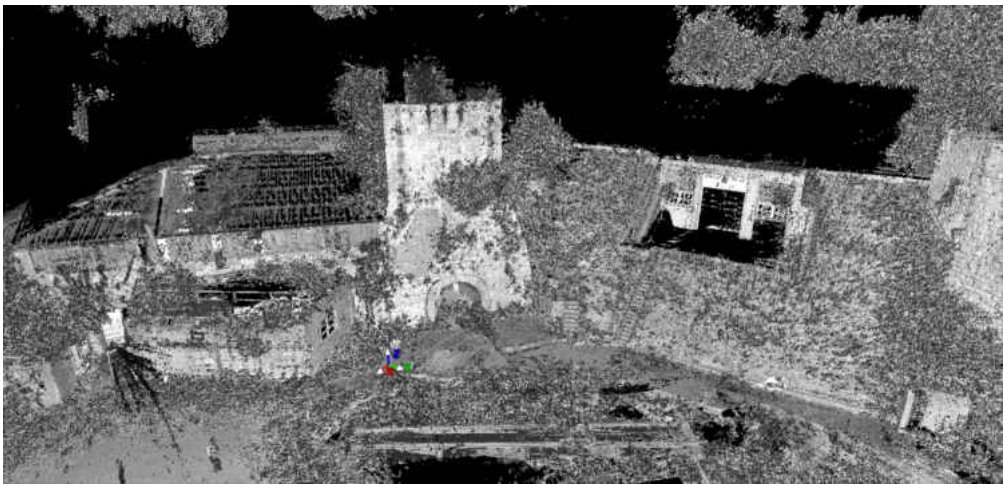


FIGURE 2.11: Brymbo ironworks, blast furnace

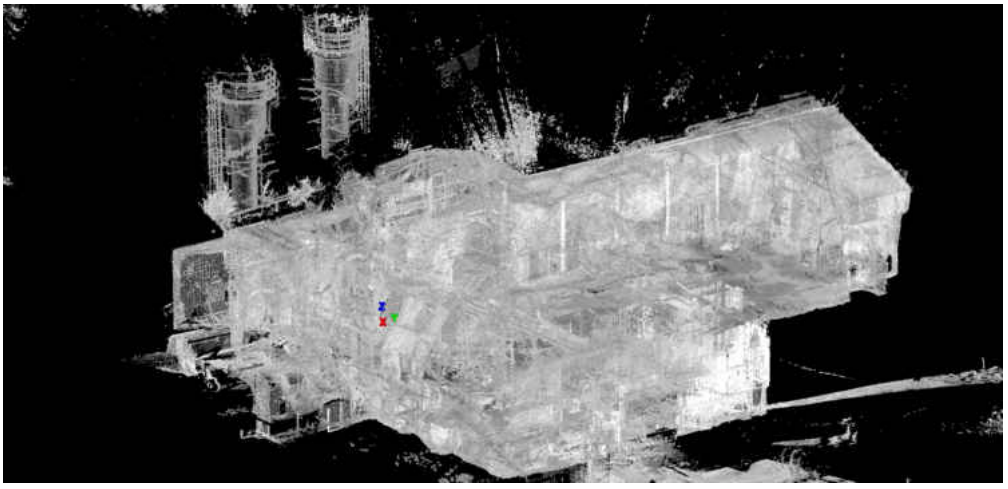


FIGURE 2.12: Brymbo ironworks foundry

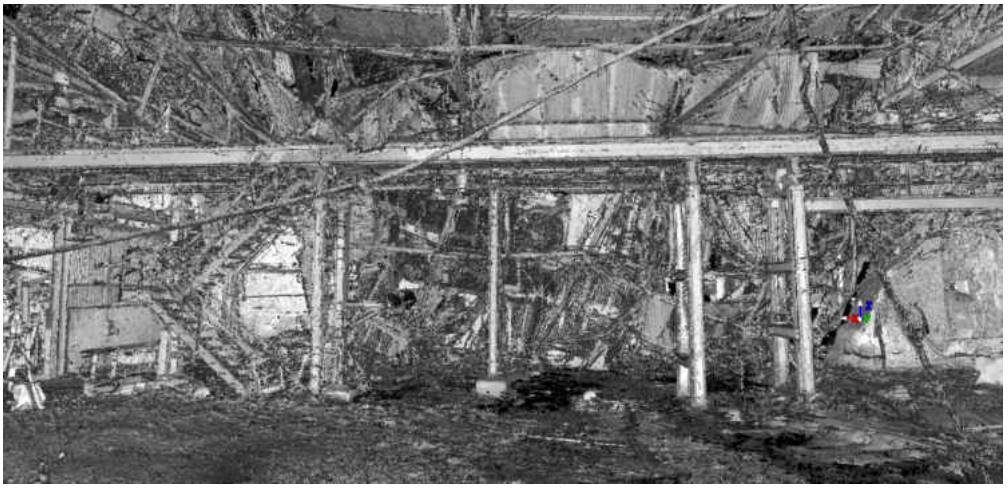


FIGURE 2.13: Brymbo ironworks foundry, collapsed roof

example picking a piece of paper from a table can be defined as three actions: identification of the table with a piece of paper on it, movement to that table and finally picking

up the piece of paper. There are many tasks humans are conditioned to perform intuitively. Looking at and identifying objects, path planning, moving and picking objects are just a few simple examples. While performing those tasks, we often mistake objects, bump into doorways and misjudge distances. Even simple tasks can be hard to define. Going back to our example, how would one identify a piece of paper? We could try by looking for a specific colour in the given scene or a specific shape. However, what is the colour of paper? White, green, blue? Is it rectangular, irregular shaped, is it folded, partially covered by something else? Humans rely on previous experience to identify objects, which can be described as concepts; we would not look for a specific piece of paper, we would try to identify any object fitting the concept of a piece of paper. It is that reliance on high level descriptions that make automation difficult. Trying to determine an accurate description of any task requires more than a set of high level instructions, it requires the description, or at least definition, of the contextual information.

This section explores the meaning of expertise in the field of terrestrial laser scanning and the issues related to capturing expertise beyond the general scanning guidelines. We are interestrd in the expertise on detection of the positions of the laser scanner to acquire the complete scan of the environment. This section only briefly touches on this topic which goes beyond the scope of this thesis. As such, this sectionshould not be treated as an exhaustive analysis of expertise acquisition.

2.2.1 Laser scanning approaches

We are interested in collecting information on the reconnaisance and position detection stages of the laser scanning process. Both stages are usually performed simultaneously by an operator, during exploration of the surroundings. After creation of a mental map of the area an operator begins to position the potential scan station. There are several approaches to scanning, used depending on the scan location and personal preferences. Following approaches are based on the English heritage guidelines [5] and experience gained in the duration of the projcet.

A first, greedy approach, relies on covering the widest possible area with a single scan, which becomes the reference for other scans. The subsequent scans are added as needed, ensuring that they at least partially overlap with the initial scan. This method is often used to scan large, open spaces both indoor and outdoor.

A second approach relies on a traversal of the area, ensuring each scan has overlap with both the former and subsequent scans. This method is often used to scan the outer elevations of buildings and when having a reference scan is impossible or impractical.

A third approach relies on scanning chambers and corridors as standalone units and ensuring the overlap between at least one pair of scans within interconnected chambers. This method is often used for scanning in cramped spaces such as caves, tombs and in large multi-chamber complexes.

These strategies lead to very simple directives that change the way an operator thinks about the consecutive scans. All three can potentially result in the same scanning positions. The main difference comes at the registration stage, during which the scans are combined to create one scene. The first approach will have the opertor use the reference scan as the origin and attempt to correlate it with all the remaining scans, leading to all scans having overlapping points with the reference scan. The second approach will rely on the correlation between the consecutive scans, where each scan is a part of two pairs. Ultimately leading to a closed loop of such pairs. Using the third approach the operator would register each of the chambers seperately, and then fing overlapping points between the adjacent, already registered, chambers.

2.2.2 Simple case scenario

The simplest case is a rectangular room, see Figure 2.14 a. Such imaginary room could be scanned in a single scan. Placing a single box in that room means that a single scan can no longer ensure the completeness of the scan. For that single box we need to add at least one more scan to ensure completeness, see Figure 2.14 b. Are two scans enough to ensure completeness of the scan assuming a single box in a rectangular room? Yes, but only under certain conditions. Figure 2.14 c and d showthat the position of the scans is as important as the number of scans. Three scans should be enough to cover a rectangular space containing at most one object, regardless of the position, or size of that object. In a case that the object contains a large amount of self-occlusions, more scans might be required to capture the object itself.

2.2.3 Increasing the complexity

Now lets look at a slightly more complex case, a room with a columnade on one side of the main chamber and an open beam ceiling. For this problem we assume that the laser scanner can only be positioned within a single plane defined by the height of the tripod. The approach is to divide the area into smaller, self contained areas (I and II) with limited occlusions and proceed as in the simple case scenario, see Figure 2.14 e. The added challenge beyond the division of the area is the open beam ceiling; we can no longer guarantee the full completeness as the scene contains areas which would be impractical to scan because of placement constraints such as the tops of the cross

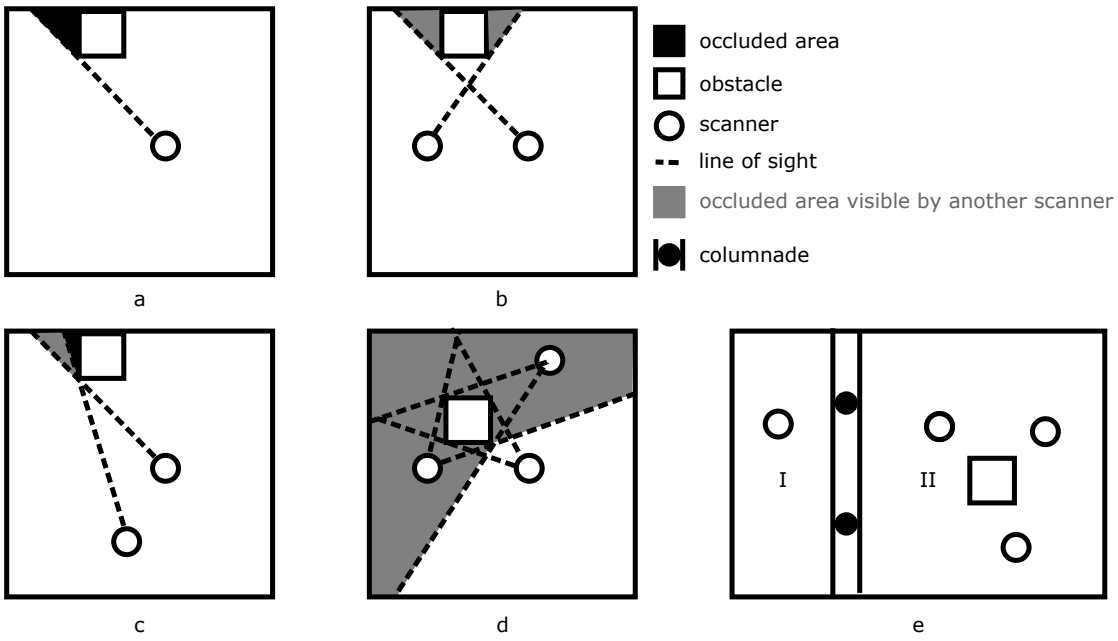


FIGURE 2.14: simple case laser scanning scenario

beams. In such a case we are considering the best possible completeness instead of full completeness. How does adding a box into this scene affect the number and position of the scans? As shown in the simple case scenario a limited number of boxes should mainly affect the positions of the laser scanners.

Given that any occlusion present in the scene can be enclosed within an imaginary bounding box, we should be able to treat the occlusions as boxes. Any area, no matter how complex, can be divided into smaller, self contained regions containing a limited number of occlusions/boxes, reducing the complex problems into the simple case.

2.3 What is an Expert?

Having described the different approaches to laser scanning it is safe to assume that anyone should be able to perform a complete scan with minimal training. Why do we need experts then? To answer this question we need to consider the contextual information about the knowledge of the operator. The main difference between the work of a novice and an expert operator can be seen in the point density of specific areas in the registered scene and the number of scans. A novice will, on average, perform more laser scans than an expert and those scans will contain larger proportion of high density areas in the low interest regions of the scene.

2.3.1 A novice

Lets imagine the laser scanner is an exposed light bulb. Anyone who can determine how many light bulbs it will take to light up the area can determine the positions of the laser scanner required to scan the area. The question is how well that task can be performed? An average person will not have any experience in positioning light bulbs and will struggle with the task. If that person consults some set of guidelines or seeks advice form others, they become a novice. A novice is a person with a limited knowledge about the task at hand. We assume that a novice is competent at following the provided guidelines and methodology. Even though a novice is capable of ensuring the completeness of a scene, due to the guidelines favouring completeness over efficiency, they usually will take more scans to do so. Another drawback is inability to estimate the quality of the scan in situ, leading to large featureless areas of high point density in the final scene. The final issue is error prevention, detection and rectification. Most novices will not be able to recognise a mistake, such as lack of sufficient overlap between scans or lack of corresponding points until the registration stage.

Characteristics of a novice:

- capable of detecting scan positions,
- capable of ensuring completeness,
- capable of following instructions and guidelines.

2.3.2 An expert

An expert is capable of estimating the quality of a scan from any given position, being able to adjust the scan precision (vertical and horizontal resolution) on the fly. Instead of blindly following the guidelines an expert is capable of determining the best course of action, even if it goes against some of the set rules. Experts tend to tweak the methodologies slightly to fit their approach to the problem better [6]. They often ignore parts of methodologies and guidelines designed to safeguard novices from making mistakes leading to the optimisation of the process.

Characteristics of an expert:

- capable of detecting scan positions,
- capable of ensuring completeness,
- capable of adjusting instructions and guidelines,

- capable of detecting errors and preventing problems in situ.

2.3.3 Becoming an expert

Becoming an expert at laser scanning is easier than, say, becoming an expert at playing an instrument or becoming a medical practitioner. The 10,000 hours rule [7] would not quite apply, as only a fraction of the time, several hundred hours, would be required to become an expert in laser scanning. The idea of requiring additional practise to perfect the skill is however at the core of acquiring the expertise. The practise based model assumes the expertise requires deliberative actions [8]. This may well apply for medical practitioners, but is it a ubiquitous characteristic of expertise? There is an alternative view, characterising expertise as an intuition driven skill acquired by experimentation to solve indeterminate problems [6]. This view subscribes to the idea of tweaking the rules to fit the current problem. Regardless of the viewpoint, expertise is an acquired skill. It is questionable if expertise can be acquired by practise alone [9, 10]. Practise does, however, play a major role in the acquisition of expertise. In case of laser scanning the practise includes the work in situ as well as later analysis of the scans and registration. It is vital to understand how scan positions relate to the generated point cloud. It can be said that every hour on location corresponds to ten hours of processing.

2.3.4 Laser scanning expertise

Regardless of the level of expertise the aim of a laser scan is to acquire the scene at the highest completeness possible. Often a reference scan is deemed obsolete and is either skipped, or performed at a very low resolution. Due to the time constraints it is however desirable to reduce the number of scans. In complex environments the number of scans is increased at the cost of resolution. The higher number of scans lessens the effect of occlusions while maintaining closer proximity to the target surfaces mitigates the lower resolution setting of the scanner. Experts agree that the best practise to ensure scan completeness is the acquisition of additional, seemingly obsolete, scans[1]. The practice improves the redundancy within the dataset. A larger number of overlapping scans ensures that the corresponding points between them can be found. The drawback of taking more scans is potential increase in the registration error. During registration the operator has to choose corresponding points between two laser scans. Such operation relies on the accuracy of the operator and any shift between those points increases error.

[1]Based on interviews with an employee of Leica Geosystems and a person who wishes to remain anonymous, but is a qualified surveyor

2.4 Can we capture expertise?

Laser scanning expertise relies on spatial awareness of the operator and the ability to estimate the quality of the resulting point cloud in situ, which is done intuitively. It is not knowledge that can be easily captured, as it relies on the subjective opinion of a human operator. The spatial analysis of the environment can be performed if we are able to model the given scene. Even if we are not capturing the exact expertise of human operators we are able to bridge the gap between the skillset of a novice and an expert. It is debatable if an automated system is capable of understanding the complexities of spatial analysis. Even if there is no understanding of the concepts involved, an engineered approach, based on the expert knowledge, is capable of replicating the results of the reasoning of an expert operator. In a sense the automated system is breaking the rules and guidelines for laser scanning to exploit its own capabilities, a behaviour characteristic of an expert.

Chapter 3

Literature review

3.1 Introduction

The project objective requires a system capable of a deeper understanding of the environment surrounding the user. Such system has to be capable of detecting places that are not visible due to occlusions. This requires a notion of visibility, which, in the heritage sector, is a binary measure signifying a line of sight between a viewpoint and an object [11]. The next requirement is an evaluation of the visibility coverage of the environment from any given viewpoint. Due to occlusions it is often impossible to ensure full coverage of a given environment using a single viewpoint. Therefore multiple viewpoints are often used to increase the coverage. Acquiring a complete scan of an environment often requires multiple partial scans; this adds the additional need for an overlap between the scans. The problem can be reduced to the need for the creation of the smallest set of viewpoints within a scene, while simultaneously maximising visibility coverage.

Most of the requirements of the described system directly correlate to the *Art gallery problem* introduced by Chvátal [12]. The first part of this chapter highlights the differences between our problem and *Art gallery problem*. The second part discusses different approaches for the reconstruction of the environment in the context of spatial analysis. The final part presents the related work. Table 3.1 presents main concepts discussed in this chapter.

Table 3.1: The overview of sections and discussed concepts

Section	Topics	Concepts
Art gallery problem	Art gallery problem	visibility within the scene visibility coverage simmilarities and differences
Environment modeling	topological map metric map	modeling of the environment 2D versus 3D maps topological and metric maps
Initial data acquisition	environment resonstruction	2D and 3D SLAM shape reconstruction methods
Spatial analysis	visibility estimation occlussion detection	spatial relations visibility regions illumination problem

3.2 Art gallery problem

The art gallery problem involves the issue of guarding an art gallery of an irregular shape. The idea is to cover the whole gallery with the least amount of guards. The solution, relying on placing the guards in the corners of the gallery, was quickly found [12], proven [13], and improved using colour coding of the vertices [14]. Despite many improvements made to the algorithms [15–19], improving the reliability and time complexity, the problem remains NP-hard, meaning it is non-deterministic within polynomial time. The placement of the viewpoints in those solutions is on the edges or vertices of the polygon, representing the gallery floor, which is an undesirable location for laser scanning. The algorithm was, in some cases, expanded to allow inconsistencies within the polygons such as holes [20, 21] or edge variations [21, 22].

A more interesting method [23], from a laser scanning point of view, relies on using the maximum cardinality of a dataset as a base, then incrementally removing data points to arrive at an optimised solution. However interesting though, the reliance on the maximum cardinality can prove impractical, considering the number of viewpoints this would generate for a real world 3D dataset. A method relying on heuristic placement of potential viewpoints was presented in [24]. A union of a set of viewpoints positioned close to the vertices and a set of viewpoints positioned in centres of mass of deconstructed convex polygons was used. This initial placement allows an efficient reduction of obsolete viewpoints. An incremental multi-agent system capable of repositioning the agents positioned near vertices can be used to improve the performance [25].

The art gallery problem does not account for obstacles within the gallery environment, which is a critical requirement for our project. There are some solutions to the expanded art gallery problem that use guards within the perimeter but away from the walls [21, 26]

or take into account the obstacles [27, 28]. Those rely on the generation of the visibility polygon [29] which is a star polygon sharing its edges with the obstacles.

The art gallery problem is very close to the problem we are facing. However solutions to it are designed for environments that can be reduced to a single 2D plane [21]. This is not true in our case as the full spectrum of the 3D environment is important. The existing solutions for the 3D art gallery problem [30] either do not guarantee the coverage of the interior space [14, 21] or are very limited by the number of vertices in the polyhedron [31].

The solutions to the art gallery problem are either limited to 2D representations or highly simplified 3D representations. Additionally the environment has to be converted to a polygon approximation. Ultimately the solutions provided are viable but might not be optimal. The time required to arrive at a solution grows exponentially with the number of vertices in a given polygon/polyhedron. We can, however reuse some techniques such as the use of multiple starting viewpoints and elimination of the obsolete ones. Last but not least, no matter what reasoning method will be used, it requires some kind of underlying representation of the environment.

3.3 Environment modeling

Environment modelling is seemingly a simple task. All it requires is some kind of representation of a given environment. There are, however, many different types of potential representations, with varying levels of abstraction, each designed with a specific task in mind. Some rely on highly conceptual level of understanding, others attempt to recreate the whole environment. The common ground is the need for a representation of a scene in a way that will support the reasoning about a given problem.

3.3.1 Topological map

Topology is a branch of mathematics dealing with properties of space that are preserved under continuous deformations [32]. In the context of mapping and environment representation it means dealing with easily recognisable positions within an environment. A topological map is a map that has been reduced to only display qualitative information, where nodes represent the key points, that are easy to recognise, and links represent the relations between said key points. In our case they indicate the ability to travel between those points. Robot navigation systems often use such representations as navigation maps [33–35]. This approach works well when the task requires the detection of

recognisable navigable points. The added benefit of topological maps is that they are fairly easy to combine [36]. The use of such maps in the project would be viable, if the best scanning positions corresponded to major, easily detectable, changes to the world representation as seen from a given viewpoint. Unfortunately many good positions are within high visibility areas which do not differ sufficiently to be recognised as key points. Additionally it is very hard to represent occlusions within a topological map. Therefore a topological map is not suitable for the reasoning about 3D space in the context of laser scanning.

3.3.2 Metric map

An alternative to a high level conceptual representation is a metric map [37]. Such a map represents the full environment in a way that preserves the distances between objects within that environment. Metric maps are used to represent objects and environments alike, at different scales. Such representation is perfectly suitable for the modelling of environments as they keep all the information about them. The drawback of representing the environment in such a way is the amount of data. This is especially true of 3D environment representations, which need to represent every object, as they grow very fast relative to the area they cover. A low resolution map mitigates that problem slightly. The resolution is however linked to scale, so as the area covered increases the resolution is usually decreased. To keep the fine detail division of large environments into separate entities might be required. Metric maps of sufficient resolution at a given scale allow accurate detection of any occlusions within the scene, leading to the possibility of estimating the visibility coverage, which is one of our main objectives.

3.3.3 Summary

To represent environments in 3D space with the ability of detecting occlusions within them requires a model retaining sufficient detail. We have chosen to use metric maps as they have the potential to retain all the necessary information for our task. There are many different ways of representing a metric map, including point clouds, volumes, meshes, etc. Each of the representations has its own drawbacks, further described in Chapter 4.

3.4 Initial data acquisition

Metric map is our chosen representation of our environments. To create a metric map we need some method capable or reconstructing the scene in 3D space. As we are attempting to detect the positions of the viewpoints for laser scanning, the process of acquisition should be quick and not necessarily as accurate as the final data set. This leaves us with the need to explore different options of building environment representations with no prior knowledge. In robotics terms the process is called mapping, and often involves simultaneous localisation within the generated map. The reconstruction step is not further developed in this thesis. This section aims to briefly illustrate the breadth of existing 3D reconstruction methods rather than provide a comprehensive overview of the field. The following are the methods we have explored as a potential part of the laser scanning automation system.

3.4.1 2D SLAM

Simultaneous localisation and mapping [38] is the conventional technique used for acquiring maps of the environment. The most popular method is the Extended Kalman Filtering based method called EKF-SLAM with its various variations, such as [39–43]. One of the ways of improving the precision of visual SLAM is increasing the field of view of the camera [44]. The use of omnidirectional cameras provides the full 360 degree field of view [45]. This method relies on Extended Kalman Filtering of the depth detection and SIFT features. The study has shown that the use of omnidirectional camera provides better orientation accuracy than conventional cameras. This provides another confirmation that vision systems work better with wide angle cameras [46].

The visual SLAM implementations often present large drift over time. To address this problem a new approach was developed [47, 48]. The combination of EKF based monoSLAM [49] with visual odometry allows for drift compensation and improves the results. The use of large amounts of landmarks between consecutive frames allows for reduced drift in relative motion estimation. This allows for high accuracy and reliability of the camera position estimation. The highly detailed landmark set from the current image is then filtered to choose a set of sparsely distributed landmarks to add to the current map. This hybrid approach combines the low computational complexity of sparse landmark 3D map with the accuracy of visual odometry. This approach shows how the strengths of one method can be used to rectify the drawbacks of another giving an overall better performance.

Some notable alternative methods have recently become more popular. Real time SLAM [50] is capable of coping with dynamic motions, such as handheld camera movements, thanks to various improvements. FastSLAM [51] relies on the conditional independence of the landmark position estimation in relation to the robot's position, alongside a combined extended Kalman filter and particle filter to reduce the computational complexity.

With the affordability of 2D range sensors new techniques are developed that attempt to map the 3D world based on range information. A new type of range based SLAM techniques were introduced. They constitute an accurate way of generating metric world representations. Many methods rely on estimating the position of the current range sensor reading within the acquired world representation. Some use rao-blackwellized particle filters [52, 53]. Others rely on detecting the orientation toward landmarks using retroreflective markers [54] or stereo vision [55]. The shape based approach to localisation proves reliable and provides and accurate measurement. This metric representation of the world scales nicely within most environments. The main benefit of the method is generation of real-time 2D metric maps of the environment that can be used for further calculations requiring precise measurements.

Other methods concentrate on the loop closure issue [56, 57]. The former is used for vision based SLAM by improving the loop closure detection using correlation between initial image and the end image. The first method uses a tree map representation of the world to accommodate different self contained parts of a larger world representation. The second uses multiple features within the scenes to compare whole scenes instead of separate features. Both methods attempt to increase the efficiency of loop closure detection for large scale environments.

Thanks to the developments in visual and range sensors, extraction of three dimensional information is becoming more common.

3.4.2 3D Slam

Two dimensional representations might be sufficient for navigation. They however lack some information for our application. To account for all the occlusions within a given environment a 3D representation is preferable. Adding the third dimension is difficult. Research in monocular 3D SLAM [58] has explored the possibilities of using wide angle camera to track sparse localisation landmarks. The described method is using weak motion modelling based on precomputed camera motion model. This approach restricts the potential movements of the camera. The 3D representation of the world is obtained by relating the localisation landmarks from the current projection onto the stored 3D representation of the world. Although this method is a great improvement over [59] it

does not scale properly and the processing time increases with the growth of the stored map. This approach shows that visual representations of the world can be transformed into a 3D localisation map. The main drawback is that the output is limited to presenting sparse localisation of landmarks in 3D space. This method lacks the necessary detail about surfaces to detect occlusions required by our project.

The popularisation of depth sensors, such as Microsoft Kinect [60], has led to new approaches to SLAM such as RGBD SLAM [61]. It relies on matching feature points within a 2D image with the corresponding points in the 3D scene acquired from a range sensor. Then it incrementally updates the global environment map with the newly acquired data. The main drawback of this method is related to the sensor itself, the range is limited to roughly 5 meters, which, in the usual applications is more than enough. However, for laser scanning in the heritage areas the ceiling of a church can often exceed this limit.

A commercial version of RGBD SLAM algorithm was implemented in a device called project Tango [62]. This toolkit is capable of mapping indoor environments and producing metric maps including colour information. Even though it seems like the perfect solution for the acquisition of the initial dataset for the purpose of viewpoint detection, it has only been released to selected developers. Therefore it was not considered for the experimental stage.

Use of range sensors does not have to be linked to images, SLAM6D [63] tackles the problem of recreating the 3D maps of the environments by registering multiple point clouds together using various iterative closest point detection techniques. This option provides fully 3D representations of the environments. However this method is computationally intensive in comparison to the other methods. Ideally we would like to use a method capable of working in real time.

Range sensors are not necessary to produce 3D metric maps. Parallax Bundle Adjustment [64] is a technique used to enhance monocular SLAM. It uses the natural parallax angle as a parameter for 3D features to generate a 3D map of the environment from images. This method provides some of the information required, however suffers from the reliance on atmospheric conditions and has limited applications indoors.

The extraction of 3D information from images leads us to another set of methods designed to reconstruct 3D environments.

3.4.3 Shape from X

Reconstruction of 3D environments does not necessarily require a depth sensor. It is possible to extract the shape of objects and whole scenes from multiple images or even a single image. Shape from X is a general term describing techniques that extract 3D information from images, thanks to different properties.

Shape from shading [65] is relying on the change in the appearance of an object due to the position of the light source. It uses the extraction of the shade gradient from the surface. This is possible under the assumption of orthographic projection and a distant light source. Later developments [66, 67] attempt to improve the method by accounting for different illumination types. The method works well in certain conditions, it is not suitable for the reconstruction of large environments with varying or diffused lighting, which are prevalent in the environments considered in our project.

Stereo Vision [68] is using the correspondence between a set of images and the position of the cameras to recreate the 3D environment. This method is very successful and led to the development of many commercial stereo cameras. The main factor in the performance is the correspondence measure. Often, scale invariant features are used for this purpose, but many other methods have been developed relying on higher level features such as shape of the object of interest [69], which are detected before the feature matching in corresponding images. Stereo vision is one of the methods considered for the initial environment reconstruction for our project.

Structure from Motion [70–72] is another method that has been developed as an expansion of stereo vision. Shape from motion relies on the datasets collected over time by estimating related camera positions using bundling methods. Structure from Motion is a reliable method for reconstruction of convex hulls in reasonable atmospheric conditions. Due to reliance on the correspondence between images it is not suitable to be used in low lighting conditions or environments with low colour variance. All in all it could be used during the initial detection for a subset of the environments considered in the project.

Shape from Focus [73–75] is another reconstruction method for 3D objects. It relies on the change in the focus at the different focal lengths. As the focal distance changes objects at a certain distance from the camera become sharper, this property is used to estimate the distance to those objects. Multiple predetermined focal distances are used and the distance is calculated based on the areas that are in focus in the given slices. Shape from Focus is a method that allows depth perception using a single viewpoint. The method does not work particularly well with large scale environments, therefore is not suitable for our purpose.

Shape from Zoom [76] relies on the optical properties of a change in the focal length during zoom. The method works similarly to shape from focus with the difference that instead of taking pictures at the predetermined focal lengths it is using the continuous response during zoom. The method is not suitable for extraction of large scale environments and is not fit for our purpose.

Shape from X provides an alternative to SLAM based reconstruction. Structure from motion and stereo vision algorithms could be used within out project in some of the target environments.

3.5 Spatial analysis

Spatial reasoning is the main objective of the project. The problem of visibility coverage estimation is similar to the aforementioned art gallery problem. The main difference in our case is the need for estimation of the coverage of the whole 3D environment. There are multiple methods challenging similar problem at a smaller scale.

One of the methods tackles the problem of visibility counting for lines in small polygons [77]. This method is subdividing the polygon into visibility areas by following the visibility transition points on a given line within that polygon, see Figure 3.1. In other words, subdivide the line of interest at any point where visibility of the scene changes and extend the corner of the edges of the scene through those points. Such implementation is good for estimation of the visible edges from any given point in a simple 2D polygon. However, it becomes increasingly cumbersome with the increase of the polygons complexity, similarly to the Art Gallery Problem. The method is therefore limited to simple 2D polygons, which is not sufficient for our application.

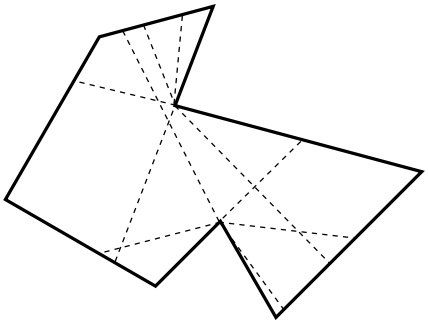


FIGURE 3.1: The visibility decomposition

A notion of spatial reasoning is very abstract, lacking standardised terms and definitions for the relations between objects. The preliminary work in this area has been presented as *Visibility Logic*, a modal logic on the spatial relations [78]. The work shows a lot

of promise for the active analysis of the surrounding environment including dynamic changes within it. The relations are described from a single viewpoint perspective and are not suitable for the estimation of scene visibility.

The most closely related work is Geometric multi-covering [79]. This paper explores the problem of illumination of an artificial, three dimensional scene. The method works by treating the viewpoints as light sources and projecting a cone of light originating from a camera onto the existing scene. The projected intensity is increased when multiple light sources overlap. The camera with most coverage of non illuminated area is selected and added to the set of cameras. The method presents several sets of predetermined camera positions, each suited for a different object. The method requires a set of potential cameras to be manually defined, as well as the minimum coverage requirement. It requires a considerable amount of human interaction and lacks a quality measure for how well an object will be illuminated/seen beyond the capability of counting how many light sources are able to illuminate it. This method works on the premise similar to ours, the main drawback is the requirement of a mesh representation and the simplistic quality measure.

3.6 Conclusions

Most visibility estimations are used to reconstruct three dimensional scenes. They allow us to match multiple viewpoints and detect overlap between them. They do however work on the premise of landmarks or key points. They are often used to detect the position and orientation of viewpoints, but they only provide basic information about the surfaces and objects. None of them provides sufficient information about the whole scene. The Art Gallery Problem attempts to solve the issue of estimating the visibility within a scene, but it is limited by solutions in the real world. Most of the solutions are impractical in the context of laser scanning. Geometrically complex scenes are considered unsolvable altogether. There is a shortage of methods used to analyse the scene, especially when it comes to multi viewpoint visibility. The existing ones either lack the quality measure, or use binary visibility metrics. Evaluation of the quality of the measure is an important requirement.

We propose a visibility measure that allows for better quality estimation, and a viewpoint detection method that works with point clouds. The proposed quality estimation relies on binary visibility of the faces of a voxel alongside the distance from the scanner and the angle of incidence between the line of sight and the center point of the voxel face. The viewpoint detection relies on the generation of the set of potential scanning position and reduction of said set to a pareto optimal solution.

Chapter 4

Reasoning about 3D data in the context of spatial perception

4.1 Introduction

This chapter explores the representation of the three dimensional environments in the context of spatial perception. A conceptual description of spatial perception will first be provided to establish the context for further fundamentals of a computational approach based on the division of representation, perception and cognition [80]. Further, the chapter will discuss the low level representation, high level data structures, and briefly discuss visualisation of three dimensional data.

Before approaching the perception and reasoning aspects, we need to establish the definition of a few terms. An *observer* is an entity capable of viewing/perceiving and interacting with the environment. The term *environment* describes a localised area, surrounding the observer, but not limited to the currently perceivable area of an observer.
Four distinct classes of environment are considered in this thesis:

- immediate surroundings: an area occupied by or within reach of the observer,
- interactable environment: an area potentially occupied or interacted with by the observer,
- perceivable environment: an area that is neither approachable nor interactable with by the observer. A position that the observer cannot physically occupy, but is still perceivable by the observers senses/sensors,

- hidden environment: an area that cannot be perceived by the observer from its current position.

To ensure complete coverage of the environment either a moving observer or use of multiple stationary observers are required. The task considered in the project is to position multiple observers within an interactable environment in such a way that the visibility coverage of the perceivable environment is maximised while minimising the number of observers.

Occlusions are objects and surfaces that obscure the line of sight of the observer, preventing him/her/it from perceiving parts of the environment. Occluded areas are volumes of space that are obstructed, creating the *hidden environment*.

Visibility coverage is the proportion of the current environment that is perceivable. Coverage is increased by reducing the extent of occluded areas.

Spatial perception is the ability to see or otherwise sense an object in the given space. From a psychological point of view it is the ability to acknowledge the existence of a feature or an object within the environment. Perception itself does not constitute understanding of the purpose or role of the features and objects. Simply being able to distinguish between a collection of background and foreground data points can be considered to be perceived as much as distinguishing between several blobs. There is no need to understand the concept of such blobs or any implications of their existence. In computer vision *spatial perception* is the ability to sense the size, shape, movement, and orientation of objects or any combination thereof [81]. In an extreme case *spatial perception* can be reduced to a on-off signal. The measurement has to be interpreted to have a meaning. Perception is therefore a basic acknowledgement of the existence of an object. There is no explicit link between the representation and the implied meaning. That implied link is at the core of *spatial reasoning*.

Spatial reasoning is an act of acknowledgement of implication of the existence of an object within a spatial context. In research performed on individuals with partially damaged brains [82, 83] it was established that spatial perception and spatial awareness are separate. Some subjects were able to acknowledge the existence of obstacles in the environment. However, they continued to bump into them. Individuals were capable of perceiving the obstacles and could name the objects they were seeing. The damaged brain was unable to correlate the knowledge of the existence of obstacles, raw data, with the need of avoiding them, intent to perform an action. Therefore an individual capable of *spatial perception* was unable to perform *spatial reasoning*. In computational terms *spatial reasoning* is the ability to correlate the perceived objects to the implications of

their existence within a given environment. Representing an environment using any internal representation mechanism is not enough, it is essential to use a representation that is capable of aiding the reasoning process. Such representation should highlight the usually implied relations between the datapoints, such as their relative position.

The rest of the chapter deals with the issues related to spatial perception. In particular it describes the low level representation of data and data structures. Also it briefly mentions data visualisation, which is outside of the scope of this thesis.

4.2 A point, a voxel, and a polygon vertex: conventional representation of 3D data

Spatial reasoning as well as spatial perception both rely on the ability to sense the environment. For humans perceiving the surroundings comes naturally. We, the observer, make assumptions about the environment based on our senses and create mental maps. For computers to be able to create mental maps or other representations of objects and environments in a spatial context, they need a way of representing positions in space. Thise representations can be very basic in terms of the internal representation, but, depending on the interpretation, can represent various relationships.

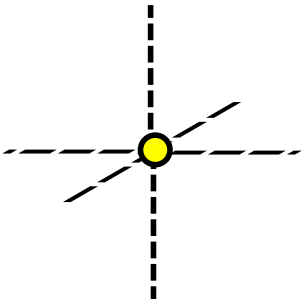


FIGURE 4.1: A 3D point

4.2.1 Point

A point is the most basic representation of a given position in space. Depending on the number of dimensions being represented, a point can contain a single or multiple values. A point is not limited to the values representing position in the given coordinate system and can contain other values, such as intensity, colour and orientation. The drawback of using a list of points is the lack of relationship between multiple points. Even though the operations on multiple points might be expensive, the list does not impede the processing

and can be converted to, or form the basis of, any other representation. This leads to the conclusion that even though a point is the most basic of representations it is also the most versatile one.

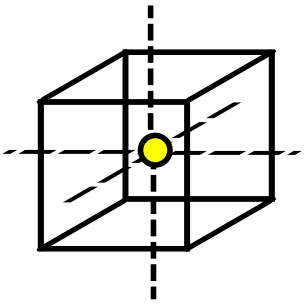


FIGURE 4.2: A voxel

4.2.2 Voxel

This is a concept similar to that of a point in a sense that in its most basic form it represents position in space. Unlike a point however, it also represents a volume of space. The properties of a voxel include its position, represented by the position of one of the corners or the center of the volume, and size, represented by the length of an edge, and relative position within an axis aligned three dimensional grid. This last property is the most important as it allows reasoning about neighbours of a voxel in a similar way to pixels. Voxels are often stored in datastructures with higher memory footprint than other lists or vertices. In its most basic form, stored in a three dimensional grid, the space requirements grow cubically in relation to the size of data bounds. This can be improved by the use of trees and sparsely populated trees, for which octrees are most commonly used, see Section 4.3. A tree based voxel representation gains the multiscale property, where the existence of a lower tree branch indicates the existence of data within the volume it represents. This property is limiting the computational requirements of iteration as large empty spaces can be easily avoided. Unfortunately, to represent the very fine detail in a large environment a vast increase in resolution is required, which leads to large dataset sizes. There is a major disadvantage to voxels: they are axis aligned, meaning that the same shape can occupy different amounts of voxels under different orientations. Also given that the space is delimited by a cube, voxels are poor at representing the orientation of features in space.

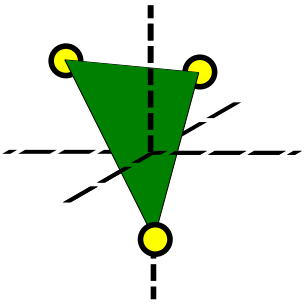


FIGURE 4.3: A polygon defined by three vertices

4.2.3 Vertex based polygons

Polygons are shapes created by connecting vertices via edges. A polygon can have an unlimited amount of vertices. However, 2–5 vertices per polygon are most common. An edge is connecting any two vertices. The inner representation of polygons is reduced to a list of vertices and edges. Such representation provides valuable additional information about the position of adjacent points on object surfaces. Such information proves invaluable when considering local relations between points or trying to determine the surface of an object. Collections of polygons, called meshes, are very good at approximating the boundary of objects. The low footprint and indication of the surface normal making it easy to detect the direction the surface is facing made polygons the most widespread representation of 3D surfaces. Additionally polygons are independent from each other and can vary in size, so a flat surface can be represented by a couple of polygons when, at the same time, an object on the top of such surface can contain thousands of polygons. This means that polygons are good at representing the tactile nature of objects, which can be described as the fine detail in relation to the size of the object. Despite those benefits, the data structure itself is not ideal for aiding reasoning about global visibility as reasoning about multiple polygons is computationally intensive. Not only the number of polygons per scene is limited, most automatically generated meshes often suffer from the surface being approximated by multiple intersecting polygons and not being watertight, a situation where the surface is not completely covered and therefore holes within the mesh exist.

4.2.4 Summary

We have described three possible data representations. An unordered list of points, albeit efficient, is hard to use in the context of spatial reasoning. A simple list allows processing of each point individually, but not in relationship with other points. To represent spatial relationships between points the dataset can either be converted to

polygons, therefore providing information about nearest neighbours of the point, or converted into a volumetric representation by delimiting the points using an equidistant grid of voxels. Polygons provide faces and ease of computation of angle of incidence for visibility estimation, however they lack spatial relations between non-neighbouring points. This makes occlusion detection and ray casting unnecessarily expensive. Voxels are very good at presenting spatial relations between points, but lack information about face direction. The volumetric representation shows only the presence of a point within an arbitrarily axis aligned space. The surfaces that are not axis aligned become jagged and as such angles of incidence are hard to acquire. All the described representations provide desirable features, however, a combination of them would be most desirable.

4.3 Data structures

4.3.1 Point cloud

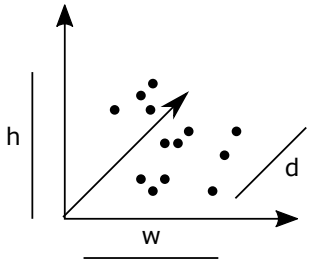


FIGURE 4.4: A point cloud

A point cloud is a collections of points in an n–dimensional space that contains a centroid and bounds for each of the dimensions. For three dimensional point clouds each point contains three values representing the positions on each of the axes, width, height, and depth. The points are represented as an unordered list, which is memory efficient to store. Point clouds are the common data structure used to store adn represent laser scans. However their simplicity makes them difficult to process in a spatial context. There are no inherent relations between the points stored in a point cloud. It is possible to compute the distribution of the points within the point cloud, however it is computationally intensive. Thanks to the negligible overhead point clouds have a very low memory footprint, only requiring the data to be loaded into memory.

4.3.2 Octree

An octree is a subdivision data structure where the parent node is divided into octants which become the leaf nodes. If there is no data in an octant the octant is treated as

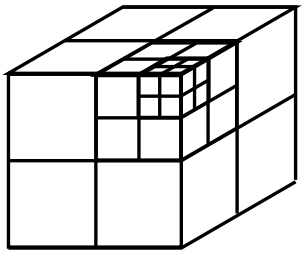


FIGURE 4.5: An octree

empty making octree a sparsly populated data structure, only increasing tree depth of the regions containing data. An octree is a tree representation of a sparsely populated grid. A full octree will require more memory than a full grid, however due to the nature of natural environments in which most of the space is empty, octrees are less memory intensive than grids. An octree size depends on the number of points within an environment as opposed to the desired leaf node resolution and the environment bounds. Thanks to the spacial subdivision and retention of spatal relations between points octrees are efficient at nearest neighbour search which allows efficient line of sight check.

4.3.3 K-d tree

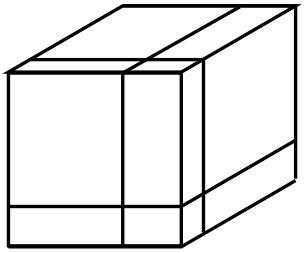


FIGURE 4.6: A K-d tree

A K-d tree is a binary tree that describes points in k dimensions. It works by dividing the current space using a hyperplane on one of the axis. The division creates half-spaces represented as child nodes in the resulting binary tree [84]. K-d trees are more memory efficient than an octree as it is a binary tree, which imcreases the minimum fill level. K-d trees provide efficient access to all the points within the data structure. However the relative positions of the points are not easily accessible. The size of the K-d tree depends solely on the number of elements stored within it. K-d trees retain the spatial relations between points, though the aspect ratio of half spaces can potentially reach extreme cases which, at times, makes access difficult. The line of sight check is efficient due to region based subdivision.

4.3.4 R-tree

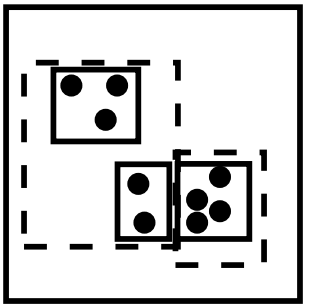


FIGURE 4.7: A 2D R-tree

An R-tree works from the premise of defining a minimum bounding box (rectangle for 2D, box for 3D) for the represented data with non-leaf nodes containing only the bounding box of the child nodes and its access identifier [85]. An overlap in the non leaf nodes is allowed, however the leaf nodes do not allow overlap between data bounding boxes. The tree level of the leaf nodes is guaranteed and the structure guarantees a minimum fill, meaning that all the leaf nodes are filled before adding another leaf node level. This is a very efficient data structure for the storage of data, allowing for easy memory paging. However the search and line of sight check are less efficient than an octree or K-d tree. It however becomes overcomplicated when extending to 3D. Being stored as quadtrees with a fixed minimum fill level R-trees have low memory footprint. Thanks to built-in memory paging integration R-trees scale gracefully.

4.3.5 Piecewise linear surface

A Piecewise linear surface is a continuous surface consisting of adjacent polygons. It is efficient in representing the surfaces of objects, retaining a good approximation of the shape and orientation of the outer surface of the objects. This data structure retains the relative surface distance between points, which allows for efficient search for nearest neighbour points contained on the same surface. Piecewise linear surfaces are not ideal for the detection of the line of sight as they require iteration through all polygons to guarantee occlusion detection. As they are stored as points and edges (list of polygons) they have a low memory footprint. The nearest neighbour detection is efficient when considering points on the surface of an object.

4.3.6 Summary

Out of the five data structures octree is chosen as the desired structure. It provides efficient line of sight detection while retaining information about spatial relations between points.

4.4 Delving deeper into a voxel core: a pseudo-hybrid data representation

The perceivable environment should be represented by a data structure that contains the information about the distance between any two points within that environment. A voxel based representation satisfies the requirement by providing fast access to voxels contained within locations of interest. However it introduces a problem due to axis alignment. Data contained within the delimited volume can be distributed in different ways, the voxel does not retain the distribution information and treats the whole volume as occupied, see Figure 4.8. This leads to otherwise smooth surfaces being represented as a jagged surface. In situations where an observer relies on the line of sight within the environment to perceive occlusions this is an undesired effect. The proposed method is relying on decomposing a voxel and is using the faces of cubes delimiting volumes.

4.4.1 Potentially partially empty space

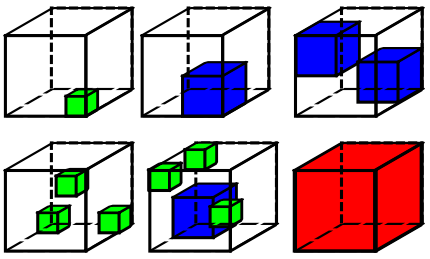


FIGURE 4.8: Possible data distribution within a voxel

The volume of space represented by a voxel can be delimited by a cube on a regular grid. Regardless of data distribution this space can be either empty, partially occupied or fully occupied. The average occupancy of a cube is very low due to the fact that most sensors are only capable of detecting the object's surface.

Similarly to the rasterisation problem [86], the creation of a voxel representation introduces many artefacts. The proposed solution is to treat the space delimited by a voxel as partially empty. Indeed we are dealing with datapoints representing surfaces of

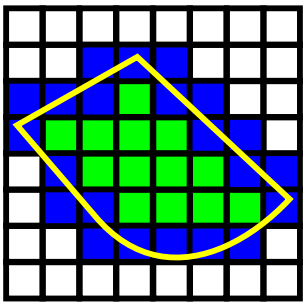


FIGURE 4.9: A yellow object boundary, represented as blue voxels

the objects, therefore it is unlikely that the whole volume delimited by a voxel will be occupied by data. The surfaces in the real world are rarely axis aligned to the same axis as the environments representation, see Figure 4.9. Most space delimited by voxels is empty, therefore the voxels are treated as partially empty, considering the whole volume delimited by a cube. This assumption carries however a rather counterintuitive computational requirement: the voxels are treated as both full and partially empty, explained in greater detail in Chapter 5.

4.4.2 Voxel face as an approximation of a directional polygon

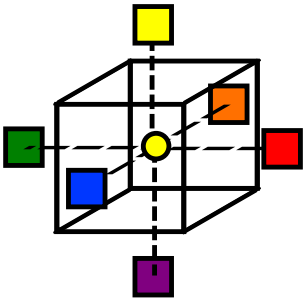


FIGURE 4.10: Proposed data representation

The axis aligned nature of a voxel is not desired. To counter it the proposed representation concentrates on the faces of that cube, treating them as separate entities. Each face is used to represent an approximate projection of the data within the delimited space onto the axis aligned plane. Such representation allows us to treat the volume of space as six polygons. Only between one to three of them are visible by an observer positioned within the interactable environment. This property is important during spatial reasoning, especially when detecting occlusions as will be seen later. Even though six polygons are represented their positions and boundaries are computed on the fly using fixed transformations, based upon the position of the centre of the cube delimiting the space. This representation provides the benefits of using voxels, allowing us to easily

relate the positions of multiple points in space as well as efficient ray casting. Similarily to polygons the faces have a direction, which allows the angle of incidence of the line of sight vector to be computed. The representation relies heavily on the properties of the voxel representation and carries most of its drawbacks, being most suitable to representing static environments. Despite its drawbacks the representation provides an efficient data structure for detecting line of sight visibility. Separation of the cube into faces allows the reasoning about the visibility from a given direction, eliminating the main drawback of using the voxel based representation.

4.5 Visualising 3D data using voxel faces

Visualisation of 3D data is always a computationally intensive process, regardless of the underlying representation. This said each representation has its benefits. Polygons are very good at approximating the tactile nature of objects as well as the surface boundaries, allowing for some advanced shading, such as ambient occlusion lighting or dynamic light source projection. Additionally each polygon has two sides allowing for directional transparency as usually the texture/shading is only applied to one side and the other is considered invisible. The main drawback is the limited amount of polygons that can be displayed at any one time. High polygon models take more memory to be loaded, even the highest detail model has fixed number of polygons. This means that the closer a camera is to the model the less detail will be visible. This is caused by the fine detail polygons within a mesh becoming larger. Sometimes the scaling issue is rectified by loading a higher resolution model as the camera zooms in. To limit the amount of polygons in most modern rendering engines such as Cryengine [87] and Unreal Engine [88] the tactile nature is being derived from normal maps rather than modelled, which adds another normal map to be loaded and increases the footprint of the rendered scene.

Volume rendering is a different approach to rendering scenes, it relies on projecting the stored data as spheres or, more commonly, cubes/voxels. Voxel based renderers are capable of rendering highly detailed environments as they rely on ray casting to find the appropriate data to display. Ray casting is the technique of casting an imaginary vector originating from the cameras focal point toward the data. Depending on application the closest voxel within line of sight or the maximum intensity value on the ray's path can be used as the display pixel. Thanks to the multiscale nature of voxels the performance is not hindered by the amount of points in the scene as drastically as polygon based systems are. The drawback of a voxel based renderer is that it portrays volumes in the same way regardless of the direction they are viewed from. Voxels have no notion of orientation, to achieve per face shading an addition of a separate UV map per voxel is

required, which due to its memory requirements, reduces the potential for directional shading. Another type of volume renderer considers all the voxels, making the objects look like semi-translucent objects with 'gel-like' appearance. These renderers rely on the data distribution and are good at displaying the relative density of data/materials.

4.5.1 Directional generalisation of pixel response

One of the main issues of voxel representation is the lack of consideration for surface texture and transparency, which can lead to sharp, jagged edges in renderings. Both properties are dependent on the direction from which the voxel is viewed, therefore inherently incompatible with a voxel representation. The proposed hybrid representation overcomes this problem by introducing directional faces. Each of the faces can repre-

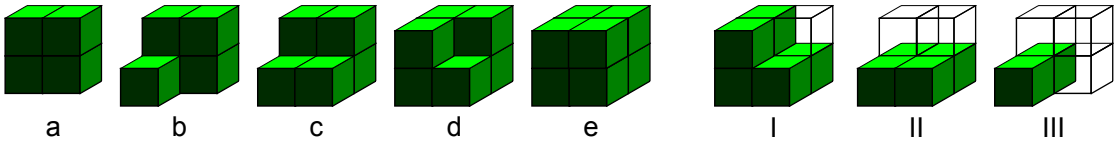


FIGURE 4.11: Surface gradient: a) low flat b) ridge c) valley d) ditch e) high flat; Transparency: I) single II) dual III) triple, of a high order cube assuming an octree implementation seen from the front

sent the transparency and other properties potentially including gradient of the surface. Figure 4.11 shows the gradient and transparency classes based on the child node distribution as seen from a viewpoint from the front. Thanks to the multiscale nature of the octree based voxel representation, with proper culling, the performance is not affected by the amount of data that is stored. The main benefit of using the hybrid representation is aggregation potential as the gradient of the surface as well as transparency can be precomputed based on the same direction faces at a lower tree level. This means that regardless of the branch depth within a tree a correct transparency value can be set for each face.

4.5.2 The close, the far, and the impractically distant.

The amount of detail is only limited by the data itself. The issue starts when we get farther away from the observer. The ray cast from the observer is forced to dig deeper into the voxel tree, thus increasing processing times. The final render is a fixed size projection onto a screen with predefined field of view. As such we can determine the size of the smallest entity to fill the pixel at any given distance. Knowing this property allows us to stop iterating through a tree when a branch level of the given size is reached. Such distance based culling allows us to reduce the time required to render a scene by

ignoring the voxels with little to no impact on the rendered pixel. Thanks to that a near-infinite resolution can be achieved, relying only on the data density.

4.6 Summary

Reasoning about spatial perception requires a data structure that will support the desired operations. We have decided to use a voxel based representation, that allows efficient ray casting and occlusion detection. Even though octrees and K-d trees provide the required functionality, octrees retain the nearest neighbour information in a more accessible way. Moreover octrees can be converted to a fixed-depth sparse representation which allows for storage of all the leaf nodes at the same tree level (similarly to an R-tree) which allows for more consistent visibility measurements. Our representation treats each face of a voxel as a separate polygon. This allows us to consider the direction from which the voxel is viewed.

Chapter 5

Scene visibility, using low resolution map as a gateway to high resolution data

5.1 Introduction

The desired outcome of the project is a semi-autonomous system that creates high resolution laser scans of given environments. The expected point density covering the areas of interest is 1mm, with the remainder of the scene at 10mm. As discussed in Chapter 3 it is somewhat a "chicken and egg" problem. To acquire the complete scan of environments, we need to map the environments. Indeed 2D metric projections are adequate in capturing the main dimensions of the environment in question. However, they fail to capture the 3D geometry of the objects within the space. Architectural features, such as arcs, beams and columns, often require more consideration than a mere plan can provide. Many objects found within the environments become additional points of interest, even though they create occlusions preventing the observer from seeing parts of the scene they are an integral part of. To capture the full geometry of the environment a 3D representation is required. Ideally, we would like to have access to a map that is as accurate as the final product and, even then, we probably would not be able to guarantee the full completeness. Due to time constraints certain occluded areas are impractical to scan and are often left out, those include the back of an object that is very close to the wall, the floor beneath the pews, etc. As it would be irrational to perform a complete full resolution scan to detect the scanner positions for such a scan, we will be considering a lower resolution dataset, the origin of which could be one of the techniques described in Chapter 3. From the very beginning we assumed the existence of a 3D metric map

of the environment making it one of the requirements of our proposed method. Our method indeed relies on the premise that a high resolution visibility coverage can be approximated by a lower resolution estimate.

Early work on this aspect of the thesis was presented in [89] and is expanded here.

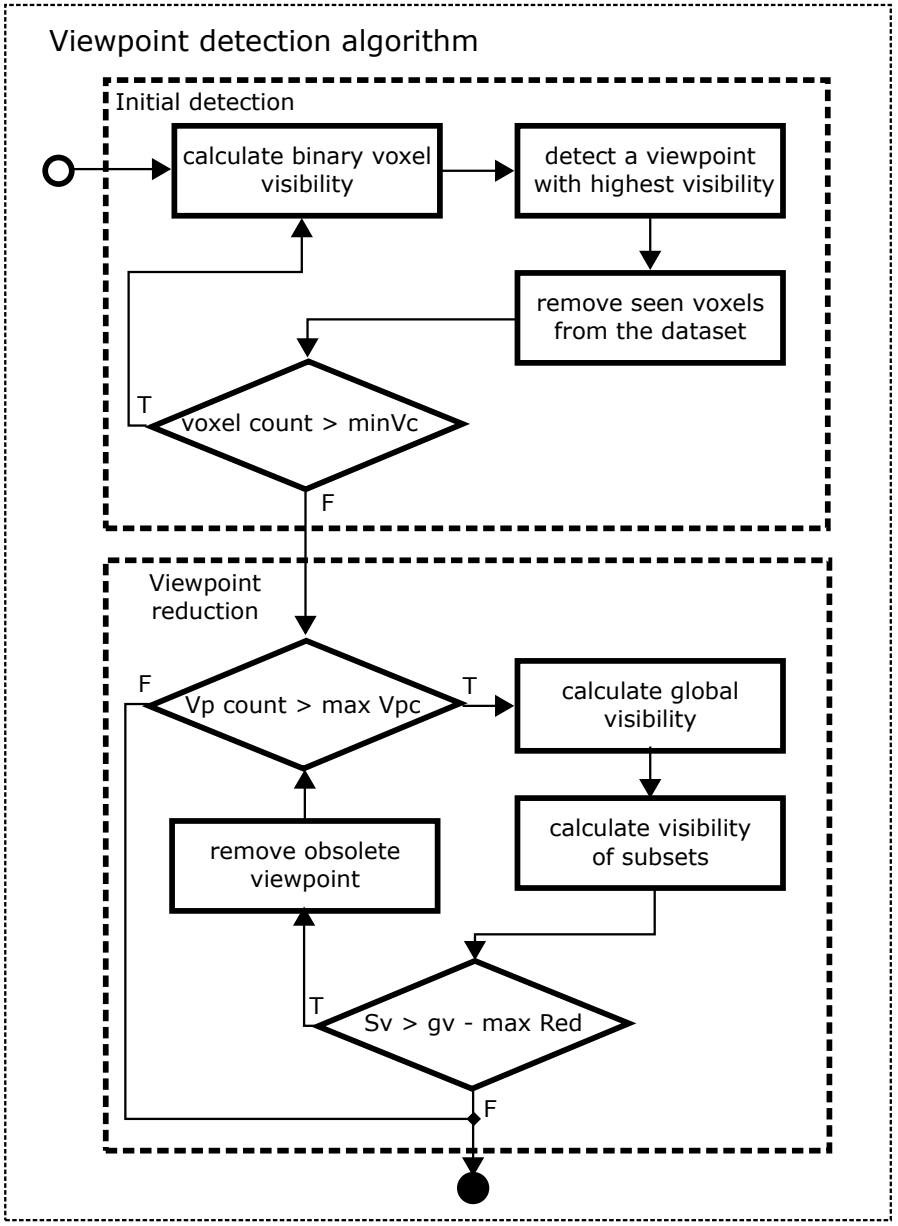


FIGURE 5.1: Viewpoint detection method overview. minVc: minimum Voxel count, Vp count: viewpoint count, max Vpc: maximum viewpoint count. Sv: subset visibility, gv: global visibility, max Red: maximum allowed reduction

This chapter describes our novel method for detection of viewpoints in a multi viewpoint visibility system. Section 5.2 introduces the visibility metric used at the viewpoint reduction stage to compare the relative visibility measure. Section 5.3 describes the methodology presented in Figure 5.1. The initial detection stage uses a simple heuristic

(A) Visible faces of a voxel in green. One, two and three visible faces.

(B) Visible inner faces of a voxel in green. One, two and three visible faces.

FIGURE 5.2: Visible voxel faces

to generate a set of potential viewpoints, then the viewpoint reduction stage evaluates this set and removes obsolete viewpoints.

5.2 Visibility estimation

Traditionally, in the Heritage sector, visibility is represented as the line of sight between an observer and another point in space [90], returning a binary result. Such representation is often sufficient for estimation of visibility from a known location. The proposed method relies on a comparison of the potential viewpoints. Therefore it requires an additional quality measure for the given visibility. A quality measure that takes into account the distance and viewing angle between the observer and the object. The presented visibility measure is using the hybrid voxel based representation of the scene, as described in Section 4.4.

5.2.1 Voxel perception

Voxels are often partially empty; such premise requires a special consideration when considering the visibility estimation. Being partially empty, voxels can be treated as both full and transparent at the same time. When a voxel is considered as full at most three of its faces can be visible. At the same time the voxel could be empty so there is the possibility for the inner faces of the voxel to be visible. Figure 5.2 shows both cases of visibility. It is important to treat the dual state of a voxel as this attempts to compensate for the overestimation of the perceived volumes.

5.2.2 Visibility potential

Thanks to the regular grid and unified voxel size there is a limited number of occlusion states between voxels viewed from a fixed viewpoint. The occlusions caused by the voxels closer to the viewpoint can cause some faces not to be visible or to be partially visible. This leads to three classes of visibility: full visibility, partial visibility and no visibility, see Figure 5.3.

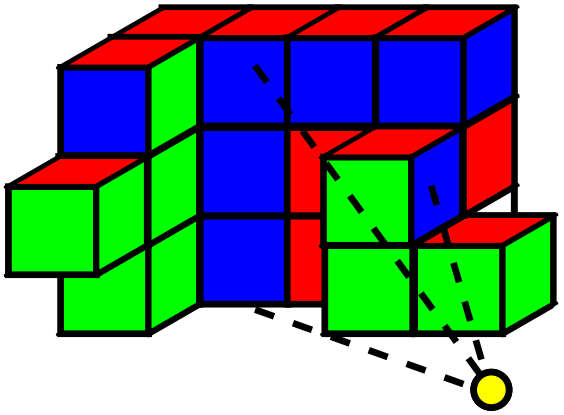


FIGURE 5.3: Example face visibility from the yellow viewpoint. Green: fully visible, Blue: partially visible, Red: not visible.

A corner is considered visible if the ray cast from the viewpoint towards the corner terminates on one of the four voxels adjacent to that corner. The approximation is caused by an undefined termination location of the raycast algorithm in this situation [91].

A face of a voxel can be classified as fully visible only when all its four corners are visible. A face is classified as partially visible if at least one corner, but not all, are visible, from the given viewpoint, see Figure 5.4. A face is not visible if no corners are visible from the given viewpoint. It can occur that a face might be classified as not visible despite it being partially visible, if its four corners are not visible, see Figure 5.4(e).

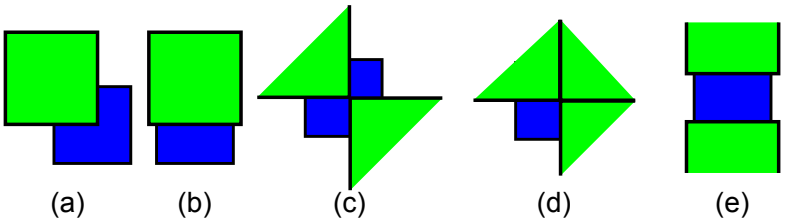


FIGURE 5.4: Partially visible faces:
(a) three (b) two (c) two (d) one (e) no visible corners.

5.2.3 Visibility quality

Visibility quality is a value ranging from 0 to 1, where 0 means not visible and 1 means perfect conditions for visibility, and it partially relies on the voxel perception classification. Partially visible face will get a value of $V = \rho$. Only fully visible faces have their visibility value computed as in Equation 5.3 where V is visibility, d and D are respectively the distance and the normalised distance between a viewpoint v_p and the center of the face f of a voxel, R_{min} is the minimum range of the data acquisition device, R_{max} is the maximum range of the data acquisition device, θ and A are respectively the

angle of incidence and normalised angle of incidence of the ray cast from the viewpoint v_p onto face f of the voxel expressed in radians:

$$D(v_p, f) = \frac{d(v_p, f) - R_{min}}{R_{max}}, \tag{5.1}$$

$$A(v_p, f) = \frac{\theta(v_p, f) - \frac{\pi}{2}}{\frac{\pi}{2}}, \tag{5.2}$$

$$V(v_p, f) = \varphi + \gamma \times D(v_p, f) + \tau \times A(v_p, f). \tag{5.3}$$

The visibility is composed of a perception classification constant (ρ for partially visible faces and φ for fully visible faces) and, in case of fully visible faces, weighted normalised distance between the viewpoint v_p and face f and a weighted normalised angle of incidence, see Figure 5.5.

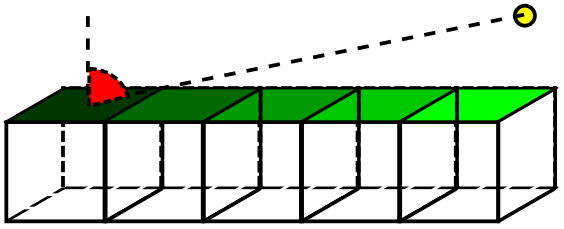


FIGURE 5.5: Visibility is affected by both distance γ and angle of incidence τ.

The chosen constants represent the significance of a given component of the visibility computation, see Table 5.1. ρ was chosen for partially visible faces to allow a small input to the visibility value. φ was chosen for full visibility as the ability to see a face plays a major role in visibility computation and is more important than distance and angle of incidence. The combined influence of distance and angle of incidence carry the same weight as the full visibility term. The distance is more important out of the pair as it influences the potential resolution. The angle of incidence is almost as important as it represents the angle the object is viewed from. This led to the assignment of weights γ for distance component and τ for angle of incidence.

TABLE 5.1: Values of perception classification and visibility constants

Constant	ρ	φ	γ	τ
Value	0.3	0.5	0.3	0.2

The method provides reasonable results even with drastically changed values, showing that it is not sensitive to the values of the constants. If ρ is set to 0 then the method ignores partially visible faces and the underestimation of the visibility value is greater.

At $\rho = 0.5$ it puts too much weight on the partially visible faces which leads to overestimation of the visibility at the edges of the projections of occlusions. φ represents the full visibility measure and is set to 0.5 to represent exactly half of the maximum visibility value. γ represents the importance of the distance from the laser scanner to the object, as the distance is directly related to the resolution of the scan this value represents the expected resolution. τ represents the angle of incidence, which is particularly important when dealing with highly reflective or absorbent surfaces. The relation between γ and τ relies on the expected distances withing the environment and the types of materials found in the scene. The values were chosen to fit an average case scenario in which the distances are more limiting than the materials.

Global visibility is defined as a normalised sum of the maximum visibilities of the faces:

$$G_v(S, F) = \sum_{f \in F} \frac{\max(V(v_p, f))}{|F|}, \forall v_p \in S. \tag{5.4}$$

Equation 5.4 describes the global visibility $G_v(S, F)$ as a sum of the maximum visibility V of a face f from any viewpoint v_p belonging to a set of all selected viewpoints S normalised by the number of faces $|F|$. F is the set of all the voxel faces in the dataset.

5.3 Multi viewpoint visibility

Our method relies on a 3D metric map as the input data. The accuracy of the method relies on the accuracy of the 3D map. To eliminate the possibility of the map quality influencing the result a decimated, unprocessed complete laser scan of the environment was used. We use a decimated laser scan as a base for generation of the volumetric representation, see Figure 5.6, to guarantee the high resolution ground truth. This is the best case scenario for the method.

5.3.1 Initial state generation

The next step is the estimation of a plane on which an acquisition device operates. Because terrestrial laser scanners are operated from a tripod a 2D plane delimiting the potential position of the scanner can be approximated. A uniform grid (1 m resolution) is then created and overlayed on the plane.

At this stage, each of the potential positions on the grid is assigned a total count of compatible voxels. A voxel is considered compatible with a position if a ray cast from

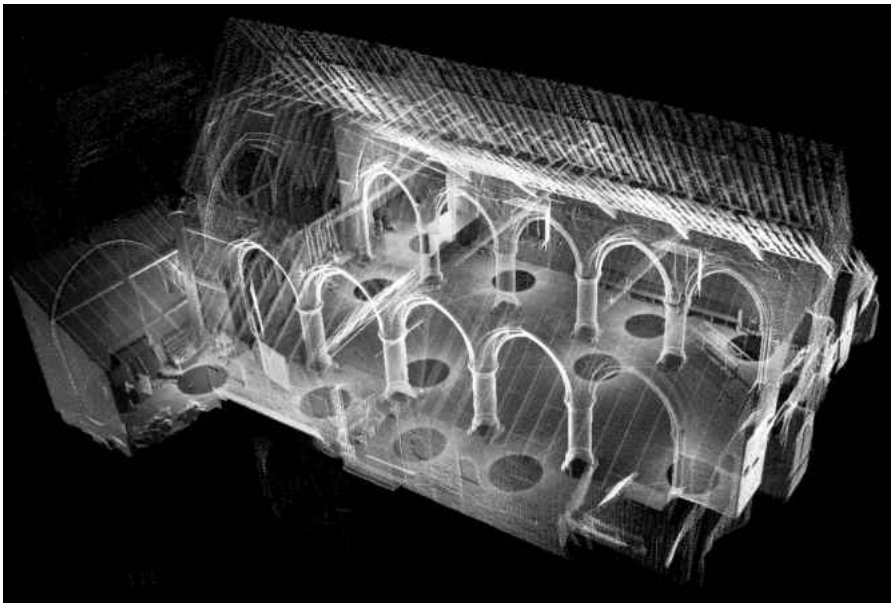


FIGURE 5.6: The laser scan of the environment used for testing

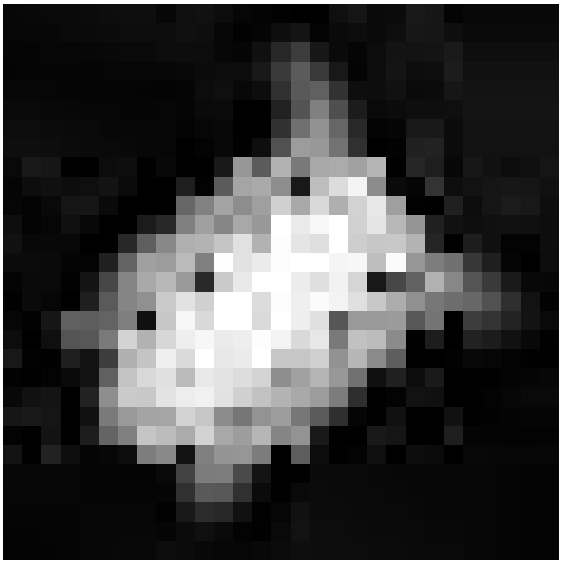
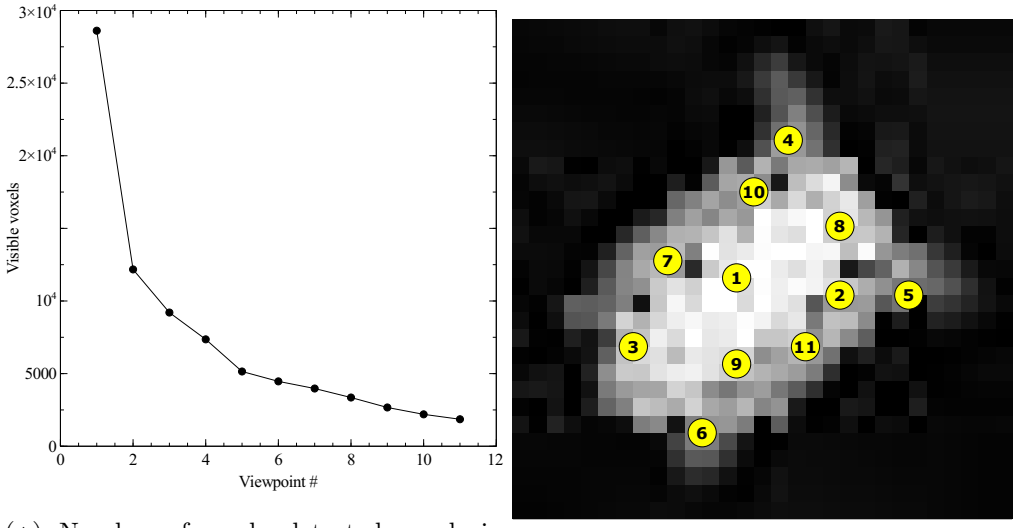


FIGURE 5.7: Voxel compatibility on the potential viewpoint position plane

the position in direction of the voxel's centre terminates on that voxel. This is a faster, though less accurate, way of estimating potential visibility of a given viewpoint than our proposed visibility measure. To eliminate the detection of the viewpoints outside of the desired area, voxels visible from the corners of the grid were excluded. Figure 5.7 presents the sum of compatibility of the voxels for each position on the 1m grid. The dark areas represent low compatibility, whereas brighter areas represent high compatibility. After the position that has the maximum number of compatible voxels has been selected, those voxels are removed from the set of all voxels. The process is then repeated until either the set of all voxels reaches 0 or the number of detected viewpoints reaches the limit of allowed viewpoints, see Figure 5.8. In the provided example a chosen limit of 11

viewpoints has been reached, only around 1% of the total voxels were left and the next viewpoint would not add sufficient amount of information to the scene. The presented heuristic is greedy and tries to maximise the amount of compatible voxels. This heuristic is fairly expensive as it relies on the creation of a grid of potential viewpoints and evaluation of each of the viewpoints. Other heuristics can be used: uniform distribution of the desired number of viewpoints within the scene, incremental placement of the viewpoint furthest from any obstacle or random placement of a moderate number of viewpoints within the scene. These heuristics will be explored in the future.



(A) Number of newly detected voxels in each viewpoint

(B) Viewpoint distribution

FIGURE 5.8: Consecutive viewpoint detection points

5.3.2 Obsolete viewpoint elimination

After a set of viewpoints have been chosen we evaluate the viability of the set as a whole. This step calculates the visibility of the set of all the chosen viewpoints and subsets of that set missing one or two viewpoints, based on the global visibility estimation, see Section 5.2. The global visibility coverage of the viewpoints in the set of potential viewpoints P excluding the set of viewpoints considered for reduction T is called a reduction potential. Global visibility estimation is computed using Equation 5.4, where $S = P - T$. Elimination of multiple viewpoints might be more beneficial than elimination of a single viewpoint during a single iteration. During a single iteration sets of two viewpoints are considered for elimination alongside the potential viewpoints.

Table 5.2 shows the reduction potential of sets permutations corresponding to one and two viewpoints within the set of potential viewpoints P. Both x and y axes signify a viewpoint number, the diagonal starting at position (1,1) shows the reduction potential of a single viewpoint, whereas other values signify two-viewpoint reduction. Viewpoint

TABLE 5.2: The reduction potential of a given set of viewpoints. Vp: chosen viewpoint

Vp	1	2	3	4	5	6	7	8	9	10	11
1	0.526	0.516	0.504	0.504	0.507	0.509	0.509	0.511	0.511	0.513	0.517
2	0.516	0.527	0.509	0.505	0.504	0.510	0.513	0.510	0.515	0.515	0.516
3	0.504	0.509	0.517	0.496	0.498	0.496	0.497	0.504	0.503	0.506	0.509
4	0.504	0.505	0.496	0.514	0.495	0.498	0.497	0.497	0.503	0.498	0.507
5	0.507	0.504	0.498	0.495	0.517	0.500	0.503	0.501	0.506	0.506	0.505
6	0.509	0.510	0.496	0.498	0.500	0.519	0.504	0.506	0.504	0.509	0.510
7	0.509	0.513	0.497	0.497	0.503	0.504	0.521	0.508	0.509	0.509	0.514
8	0.511	0.510	0.504	0.497	0.501	0.506	0.508	0.523	0.511	0.511	0.514
9	0.511	0.515	0.503	0.503	0.506	0.504	0.509	0.511	0.525	0.514	0.515
10	0.513	0.515	0.506	0.498	0.506	0.509	0.509	0.511	0.514	0.525	0.517
11	0.517	0.516	0.509	0.507	0.505	0.510	0.514	0.514	0.515	0.517	0.528

11 has the highest reduction potential, closely followed by viewpoint 2 and viewpoint 1. In the example none of the viewpoints are obsolete, therefore none of them will be removed. This method is designed to provide a pareto-optimal solution with lowest data loss during viewpoint elimination. The system can be configured to perform viewpoint elimination until a maximum allowed number of viewpoints is reached.

5.4 Experiments

The following experiments were performed using real world datasets, acquired as part of the project. All scanner positions used by a human operator were placed in accordance to English Heritage guidelines [5, 90, 92] and validated on site by an expert from the Royal Commission on the Ancient and Historical Monuments of Wales. The proof of concept experiments were performed on artificially generated shapes containing high occlusion count and rapidly advanced to the real world data stage. The presented experiments were performed on the following datasets: St. Nicholas church in Grosmont, see Figure 5.6; St. Marcella's church in Denbigh, see Figure 2.8; and Yr Hen Gapel, a chapel in Llwynrhydowen, see Figure 6.7.

5.4.1 Evaluation

The evaluation of the method is not straight forward. There is no ground truth that could be used to compare the results to. The comparison of low resolution results with the high resolution data seems to be as close to ground truth as we can get. The problem is that the tested algorithm is used to generate the ground truth. This does not mean that the results should not be compared to other resolution data. The main evaluation is the comparison of local and global visibility across multiple resolutions.

The expected result is a gradual increase in the visibility as the resolution changes, due to less data being occluded by the large voxels. The second evaluation method is a visual inspection of the visibility of a completed set of viewpoints as well as a single viewpoint. This works well with a 3D model, but does not provide satisfactory results as rendered stills. The last evaluation method is the comparison of the coverage between a set of positions chosen by a human operator and the automated system. This is the least reliable evaluation method as multiple solutions will return satisfactory results.

5.4.2 Single viewpoint visibility comparison across multiple resolutions

The tests compare multiple representations of the same dataset at different resolutions: 50cm, 20cm, 10cm, 5cm and 1cm. Table 5.3 shows the results for individual viewpoints detected by the system at each of the tested resolutions. As expected the visibility is increasing as the resolution increases. The algorithm is consistently giving a slight underestimate of the visibility estimate value from the same viewpoint at a higher resolution.

Viewpoint	1	2	3	4	5	6	7	8	9	10	11
50cm	0.15	0.09	0.13	0.06	0.05	0.08	0.08	0.14	0.12	0.07	0.08
20cm	0.19	0.14	0.16	0.08	0.07	0.11	0.12	0.16	0.17	0.11	0.12
10cm	0.22	0.17	0.19	0.11	0.1	0.13	0.15	0.19	0.2	0.14	0.15
5cm	0.26	0.2	0.22	0.13	0.13	0.17	0.19	0.23	0.23	0.18	0.19
1cm	0.3	0.26	0.26	0.17	0.16	0.2	0.22	0.27	0.27	0.22	0.23

TABLE 5.3: Visibility from given viewpoint at a given resolution

The quality of the visibility decreases with the increase in resolution due to the higher distance to the visible voxels, but remains in the range of 0.63-0.7 across the resolutions.

5.4.3 Comparison to human operator

Another evaluation step is the comparison of automatically detected viewpoints to the viewpoints used to capture the laser data in-situ. Both direct position comparison and quantitative evaluation were performed. The direct comparison shows some overlap between the selected positions, see Figure 5.9. The human operator has selected the positions in a more geometric fashion, keeping the viewpoints in line, whereas the automated system resulted in a more chaotic distribution. Both however have decided to put a scanner by the corner of an alcove, as well as in the highest visibility area.

The human detection obtains a visibility value of 0.53716 against the automated system at 0.53553 meaning that both solutions are considered equally good according to our

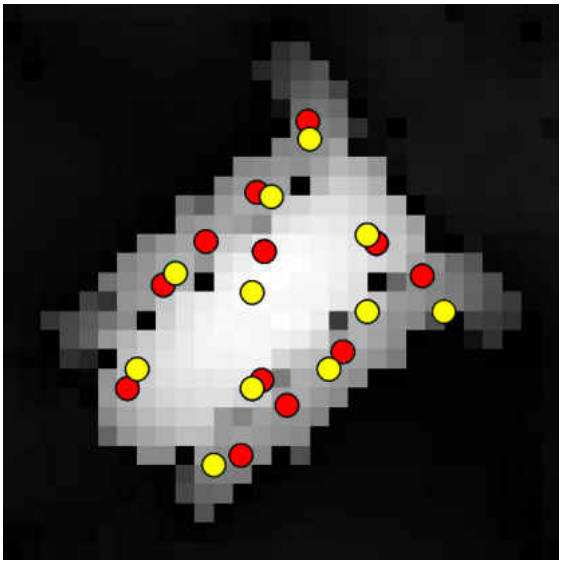


FIGURE 5.9: The detected viewpoints. Human operator in red, automated system in yellow

Detection type	Visibility	Fully visible faces	Partially visible faces
Our method	0.53553	0.479082	0.4017
Human operator	0.53716	0.480235	0.404725

TABLE 5.4: Global visibility of Grosmont dataset

visibility evaluation, see Table 5.4. The notable difference is the use of 12 viewpoints by the human operator and 11 by the automated system. The visual comparison is presented in Figure 5.9. The human detection was performed by the author on location, following the guidelines for laser scanning [5]. At the time the author was an intermediate user and performed additional scans to ensure completeness. Those additional scans were not counted toward the final output as they were redundant. The 12 best viewpoints chosen for the comparison were the 12 required scans (out of 15) based on the expertise acquired during the rest of the project and correspond to what an expert operator would have chosen as scanning positions.

5.4.4 Denbigh dataset

The Denbigh dataset corresponds to a complex indoor environment with multiple dividers. The multitude of objects placed within this scene makes it even more interesting. The method generates 11 points of interest and removes one, see Figure 5.10. The resulting set of points provides a solution that is comparable to the human operator's solution.

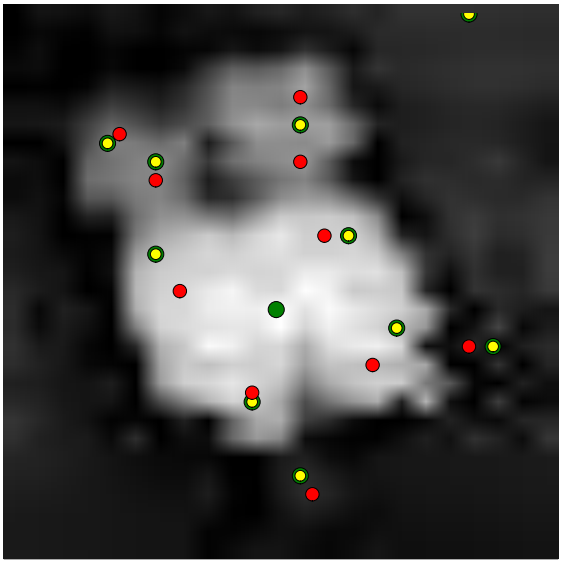


FIGURE 5.10: Detected viewpoints in the Denbigh dataset: green circles show detected viewpoints, yellow show selected viewpoints and red circles show the choice of a human operator

Detection type	Visibility	Fully visible faces	Partially visible faces
Our method	0.334326	0.273391	0.347208
Human operator	0.302817	0.245867	0.312316

TABLE 5.5: Global visibility for the Denbigh dataset

Table 5.5 shows that our method has higher visibility, however, the higher visibility is partially due to a point outside of the area being chose by the automated system. . This could be rectified by the use of an area of interest.

5.4.5 Hen Gapel dataset

The Hen Gapel dataset corresponds to a chapel on a crossroad, surrounded by buildings. A fence provides a barrier on two sides of the perimeter. The method generates 13 points of interest and removes 8, see Figure 5.11. The resulting set of points provide a solution that is comparable with the human operator's solution.

Detection type	Visibility	Fully visible faces	Partially visible faces
Our method	0.124025	0.0906868	0.174342
Human operator	0.108122	0.0745421	0.164957

TABLE 5.6: Global visibility for Hen Gapel dataset

Table 5.6 shows that our method has higher visibility. Our method provides a slightly higher estimate than the solution provided by a human operator. Both solutions are equally viable in the real world.

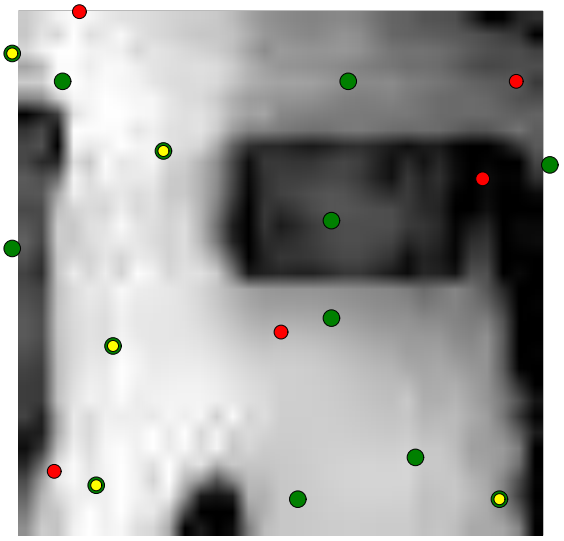


FIGURE 5.11: Detected viewpoints in the Hen Gapel dataset: green circles show detected viewpoints , yellow show selected viewpoints and red circles show the choice of a human operator

5.4.6 Summary

We have presented a method capable of detecting positions of the laser scanner. Even though parts of the method are computationally expensive, it is capable of producing solutions on par with a human operator. There is a need for the improvement of the performance as well as addition of fail safes. Our method does cope well with complex environments and multiple occlusions. The visibility quality measure provides the much needed expansion to the state of the art visibility measures.

Chapter 6

Managing large datasets

6.1 Introduction

One of the outcomes of the project is the generation of high resolution scans of various environments. Those scans end up stored as point clouds, that is lists of points representing positions in 3D space, see Chapter 4. Point clouds are reaching thousands of millions of points. These unwieldy behemoths are often intimidating to the newcomers and experts in the field alike. Most users tend to give up on using them after a couple of unsuccessful attempts to load them. This can be rectified by development and distribution of support tools that can help the users to automate the processing of the data. Point clouds are used as sources for virtual survey, converted to digital elevation maps or used to map the environments. The current 3D modeling software packages, such as Maya [93], Autocad [94] and ArcGIS [95] are limited to loading several million points at a time. This vastly limits the potential user base. To ensure the usability of point clouds it is essential to provide ways of using them with little to no prior experience.

The most accepted option is to provide a way of loading point clouds, or their subset, to the commonly used 3D modelling software. The conventional methods of handling point clouds include chopping them into often unmanageable pieces, which often need to be further subdivided, or decimating them.

Decimating a point cloud often leads to losing vast amounts of potentially useful data. The loss is unavoidable, but in certain conditions the severity of the loss can be reduced. The most common method of decimating a point cloud is naive decimation, where only every n^{th} point is left in. The second most popular, yet expensive, is volumetric decimation where an octree is created prior to decimation and the centres of the created leaf nodes are returned as the new points. The proposed method is designed with the processing of large point clouds in mind and works on a point by point basis.

This chapter describes a new method for volumetric decimation that does not require the creation of an octree during the decimation process. Instead it relies on computing the position each point would occupy in the volume at a desired point density. Section 6.2 briefly introduces the existing decimation methods. Section 6.3 describes the methodology and different implementations of the method. Section 6.4 provides a visual and performance comparison between the every n^{th} point decimation and our method. The comparison is mostly illustrative as the direct comparison with a volumetric decimation method was not possible due to the size of the target datasets.

6.2 Common reduction methods

Different decimation properties are required when trying to detect an object within a scene, attempting to combine multiple point clouds together or using the point cloud as a reference for reconstruction. The properties in question are:

- point density retention,
- point density normalisation,
- reduction by point ratio/interval,
- reduction by point density,
- noise retention,
- surface geometry retention.

The point cloud reduction algorithms perform anything from simple decimation to complex surface curvature retention. Reduction ratios upwards of 90% are not uncommon. This overview will only mention several algorithms representative of various complexity levels.

The most basic decimation method decimates every n^{th} element. In this case the user provides either the reduction ratio or the reduction interval and the algorithm iterates through the list of points and saves every n^{th} point. The algorithm is simple and reliable with the computational complexity of $O(N)$. The benefits include low memory footprint since only one point is loaded at a time and low computational requirements since each point is iterated only once. The method is however highly dependent on the order of points within the original point cloud. In the worst case scenario returns a cluster of data containing only a fraction of the dataset at initial, pre-decimation, density and eliminates most valuable points. The worst case scenario is however unlikely

or borderline impossible without special preparation of the data. The average case will return a random subset of the original point cloud that mostly retains the relative point density of the original. The resulting dataset is prone to being noisy as the algorithm has no notion of discerning noise.

The second common reduction method is volumetric decimation. The user provides a tree level or a leaf node size, to determine the size of a volume delimited by voxels which correspond to the average point density. The algorithm then creates a tree based representation with the time complexity of $O(N \log N)$ and saves the centres of the voxels containing data as final points. The memory footprint is moderately high requiring an octree representation of the whole dataset. For example, a dataset containing 512 points would require an overhead of 584 to 4680 pointers. The computational requirements are moderately low since each point is iterated once and a single traversal of the resulting octree is required. The main drawback is that the resulting point cloud does not retain any of the original data and the surfaces are often shifted, sometimes leading to the creation of parallel surfaces due to the axis aligned nature of the voxels. The resulting dataset has fixed point density across the whole scene. The algorithm tends to amplify noise due to the retention of the noise points and equalisation of the point density. This can be mitigated by the use of density analysis to detect the centroid of all the points belonging to a volume.

Point Cloud Library [96] implements an alternative version of the volumetric decimation algorithm. This implementation reduces the diffusion effect and improves the general quality of the resulting point cloud at the cost of increasing both memory requirement and computational complexity. All of the points within the dataset have to be loaded to the memory. The method relies on computing the centroids of all the leaf nodes, requiring an additional iteration through all the points. The main disadvantage is that the maximum point cloud size that the algorithm can cope with is vastly reduced, the actual size depending on the amount of memory available on the processing machine.

Some of the more complex algorithms, such as the global clustering with geometrical retention [97] and point sampling of the surface [98] rely on clustering of the points for the decimation. The user selects the desired number of points and the algorithm attempts to fit all the surface points within the number of clusters equal to the desired number of points. A cluster is a collection of nearest neighbour points to the centroid of the cluster. The memory footprint as well as computational complexity are high, both methods requiring the whole dataset to be loaded into memory and iteratively cluster the nearest neighbours until the desired number of clusters remains. With a tree representation for the data the memory requirements are between double and 10 times the original dataset size, which is further doubled by the clusters themselves. Multiple iterations over the

dataset lead to high requirements for processing even moderately large datasets. The advantage of using cluster based reduction is high surface smoothness. Even though the resulting point cloud does not retain the original data it approximates the shape of the original in a well behaved manner. One of the drawbacks is the disparity between the geometry of the original and the resulting surface. This issue can be mitigated by the use of geometry disparity measure during the clustering process [97]. The resulting datasets tend to retain the relative point density of the scene.

Another algorithm [99] relies on re-composition of the model from the initial data incrementally adding points that are furthest away from any data point currently retained. The user either provides the desired point density, a feature satisfied by a cartesian grid analysis, or the desired point count which is satisfied by the incremental nature of the algorithm. The algorithm then creates a cartesian grid holder for the resulting set alongside a maximum distance heap and then populates the grid incrementally with the time complexity of $O(N \log N)$. The memory footprint is high due to the creation of a cartesian grid for the output as well as the storage of maximum distance heap and the computation complexity is moderate. The main drawback is the high memory requirement, limiting the maximum size of the point cloud. The resulting point cloud has mostly uniform point distribution across the surface as well as a normalised point density across the scene. The algorithm is susceptible to noise retention as it relies on the furthest distance points.

The main trade-offs involve memory usage versus the quality of the resulting point cloud. Most complex methods rely on the statistical analysis of the neighbouring points and require the whole dataset to be loaded into memory. In contrast, the simple method provides a fast and memory efficient way of obtaining a decimated point cloud.

The proposed method is emulating the principle of the volumetric decimation, without the requirement of spatial analysis of the data, which allows the algorithm to process the point cloud on a point by point basis. Being able to replicate the result of the volumetric decimation without the need to load the whole dataset into the memory would make the method more appealing to the users.

6.3 Proposed point cloud reduction method

Despite the multitude of the point cloud reduction techniques, there seems to be a lack of memory efficient methods capable of producing a fixed point density result without the need for the spatial analysis of point neighbourhoods. This led to the development

of such a method, inspired by the knowledge representation techniques used in artificial intelligence. The aim was to create a conceptual model which is not influenced by the potential implementation. Such technique can then be adapted at the implementation stage to trade-off time versus memory requirements. Our method is capable of directly replicating the result of a conventional volumetric decimation, with the memory requirement linked to the output size, not the size of the dataset being decimated.

6.3.1 General overview

The user is required to provide the desired point density by specifying the size of a voxel that is supposed to contain a single data point in the decimated point cloud.

The algorithm works on a four step framework:

- Read,
- Process,
- Evaluate,
- Resolve.

The *read* step is iterating through the point cloud, loading one data point to the memory at a time. This has the advantage of reducing the memory footprint by removing its reliance on the original data size.

The *processing* step generates a hash code unique to the projection of the actual position onto a search space with the help of the hashing function in Equation 6.2. This step ensures that only the position information is used in the next step.

The *evaluation* step checks if the positional value is present in the current knowledge base. This process is imitating the awareness of the existence of the data within that position.

The *resolution* step decides whether to discard the data point or keep the information about the occupancy of the new space and add the original data point to the reduced point cloud.

Figure 6.1 shows the the steps of the algorithm.

The main benefit of this design is that the memory requirement is directly shifted to the implementation of the knowledge base and the hash code. Another difference to the standard volumetric decimation that uses the centres of the voxels as new points is that

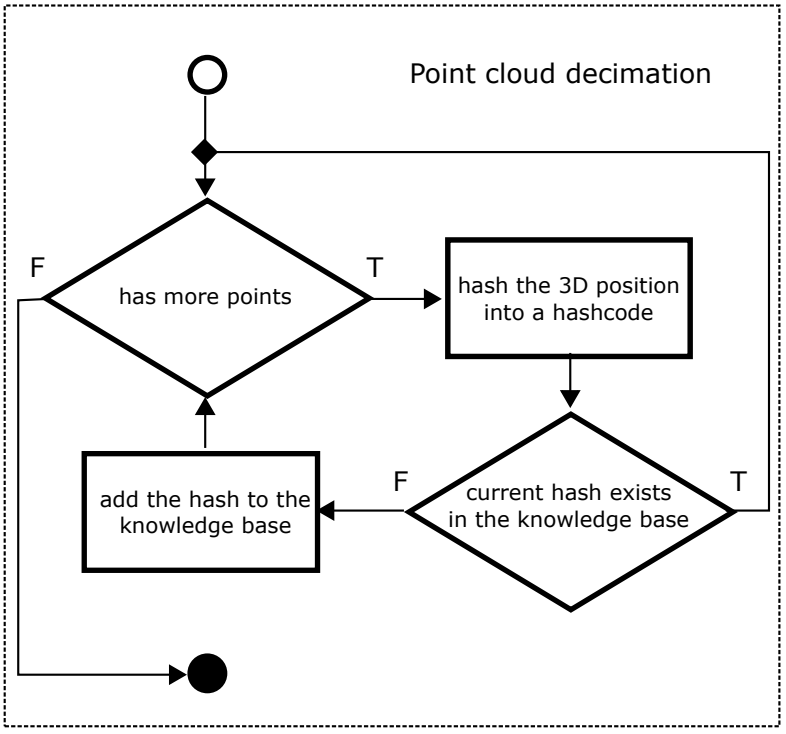


FIGURE 6.1: Point cloud decimation algorithm

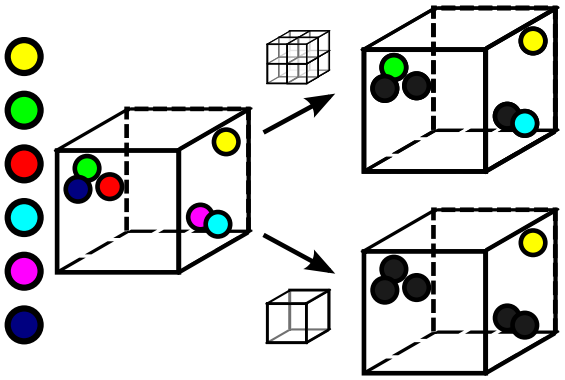


FIGURE 6.2: Example of first point choice operation

the proposed method uses the first point belonging to the volume. Figure 6.2 shows the implications of choosing the first point as the representative value of the volume, regardless of other points belonging to the volume at different scales. Thanks to the assumption that the first encountered point is representative of a volume the method is able to, on average, reduce the potential surface deformation in comparison to the use of the centre of a voxel. Using the centre of a voxel would return a central point regardless of the data distribution within the voxel. What is more, it would alter the data itself, changing the structure of the surface. Using the first encountered element could potentially lead to the retention of noise instead of valid data, but on average it will retain mostly surface points and, what is more important, keeps the original data, so it is not altering the surface structure as much as the alternative.

6.3.2 Search space dimensionality reduction

Working with 3D datasets can be cumbersome, especially when considering the spatial analysis aspects of the task. The data has to be loaded into special data structures and then processed as a whole to achieve the desired result. The resulting processing has to deal with the fairly expensive point acquisition from the data structure just to get the data it will be processing. This requires multiple iterations through a list of points or creation of supporting data structures. In the field of artificial intelligence a common practice is to convert the processing problem into simpler search problems, limiting the search space. A search space is a set of all the possible states of a problem, or in our case all the points in the decimated point cloud. Knowing that we will be detecting the volumes occupied by the data we can convert the measurement precision, precision with which data points describe the 3D position, to match the positions of the desired volumes instead of actual 3D coordinates of the point. This process is further explained in Section 6.3.3. Having reduced the spatial analysis of the data to a simple coordinate conversion we can tackle the issue of a three dimensional search space. If we ignore the meaning of separate values and treat the data as one continuous stream we can reduce the search space to a single dimensional problem. This reduction of the dimensionality of the search space can potentially reduce the average time complexity of the single point access to $O(1)$ instead of $O(\log N)$ for the three dimensional search space. This means that the complexity of the algorithm can, depending on the implementation, be reduced to anything between $O(N)$ and $O(NM)$ where M is the number of points in the resulting decimated point cloud.

6.3.3 Hashing the data

The hash function is responsible for the transformation of X, Y and Z coordinates into the position of a voxel it would occupy alongside the creation of a single stream of data representing said position. The size of a voxel is defined by γ, which corresponds to the desired point density represented by 1 pixel per γ^3. For example a desired density of 1 pixel per cm^3 would result in 1cm edge length of the voxel. Equation 6.1 represents the coordinate reduction where h is the reduction function and d is the data. Equation 6.2 represents the creation of a single stream of data, reducing the search space to a single dimension with H representing the hash function. For example a point $x = 2.6543, y = 4.5436, z = 0.0325$ with voxel size $\gamma = 0.1$ would result in a hash $''26\_45\_00''$.

$$h(d) = \left\lfloor \frac{d}{\gamma} \right\rfloor, \tag{6.1}$$

$$H(X, Y, Z) = "h(X)\_h(Y)\_h(Z)". \tag{6.2}$$

The hash is represented as an immutable string, with the positions being represented in a character form. From now on the components of the hash will be treated as separate entities with the order of occurrence being the important factor. In our tests we use a hash function allowing cube size changing by one order of magnitude, with sizes of 1m, 10cm, 1cm, 1mm, 0.1mm, etc. This factor allows for sufficient decrease in resolution while ensuring meaningful change in spatial resolution.

6.3.4 Knowledge base

The knowledge base is the main component that is responsible for elimination of unwanted data points. Its primary function is storage and comparison of the hash codes. Each of the components of the hash code is evaluated against the current knowledge state. Figure 6.3 presents an example addition of a new, unknown, state to the knowledge base as well as the rejection of a previously seen one. Lets assume that we have data composed of coloured shapes, but we are only interested in having a set of unique shapes. In this case we could collect all of the coloured shapes, sort them by shape and choose one from each group of shapes. Alternatively we can use the knowledge of our desired outcome to choose the correct shapes as we see them for the first time. If we prepare a suitable location for our collection of unique shapes we can add the missing shapes from our dataset and discard the shapes that we have already collected. The colour of a shape in this case becomes irrelevant.

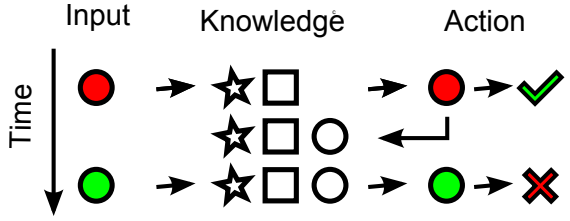


FIGURE 6.3: Knowledge base functionality. A new element, red circle is evaluated against the knowledge base. It is hashed as a circle, which has not been evaluated so far. The knowledge base adds the notion of a circle to its database and returns a confirmation of acquisition of a previously unknown state. The next time a different circle element is being evaluated the element ends up discarded.

We can apply the same principle to the positions. If we consider the locations in our desired space, all we have to do is to check if we already have seen a point occupying the specific coordinates in our desired space, the actual position of the point in the initial dataset becomes irrelevant.

A suitable data structure will provide fast check if the hash code is already present in the data structure and fast addition of an element to the data structure. Additionally it has to deal with sparse data distributions. Data structures such as sorted lists and hashes seem viable. However a sorted array takes up to $O(N)$ per data check and a hash requires a moderately large amount of memory.

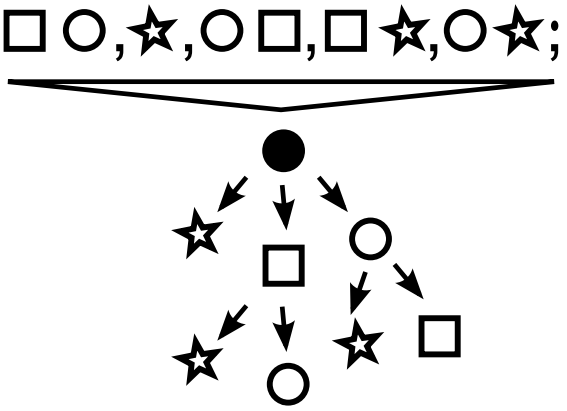


FIGURE 6.4: Knowledge base represented using an n-ary tree. Data streams stored in the tree are shown at the top. The black dot represents the root node containing branches for each of the unique first elements of a data stream. Similarly the second level contains branches leading to the unique secondary elements.

Each hash code is represented as a stream of codes, denoted by characters, arranged in a specific order. The length of the stream relies on the physical data values as well as the desired resolution. It is variable between streams. The system could instead represent the pool of all possible hash codes as a sparse n-ary tree. Figure 6.4 shows how a list of data streams is represented in a tree structure. The sparse tree will allow to use less memory while populating the knowledge base. Additionally, both the addition and occurrence check operations take at most m operations, where m is the length of the hash code.

Sparse tree representation in a n-ary array, where n represents the number of glyphs in the hash code, provides a representation that performs adequately, however the memory footprint remains fairly large due to the high depth of the leaf nodes. In a 4 decimal point precision the leaf nodes are expected at level 21 of the tree depth. This results in the memory requirement of an order of 10 times the size of the resulting decimated point cloud. Each check takes on average 60 operations.

An ordered list implementation only uses about 3–4 times the size of the resulting decimated dataset and with different search techniques provides around 60 operations per check on a dataset containing 100 million points, which is still on average lower than the tree representation. The issue with a list representation is a high insertion time (requiring the shift of all the elements after the inserted element). A linked list

would introduce at least N additional pointers, where N is the number of points in the resulting decimated point cloud.

The most efficient data structure seems to be a hash map as it only reorders itself when it needs to grow. The memory requirement of a hash map falls between a list and the tree representation. Number of operations per check is in the order of 10, relying only on a hashing function and the conflict resolution algorithm it is best performing method in an average case scenario.

6.4 Experiments

The main experiments use a non optimised implementation of the algorithm in the *Perl* scripting language. The knowledge base implementation is done using the hash data structure and the hashing function is implemented using regular expressions.

The tests compared the run times for the datasets established on an intel pentium i7 laptop with 8GB of RAM under the Windows 7$^{\text{TM}}$operating system. The test results performed on the same machine under Ubuntu 13.04 linux were comparable and will not be presented.

6.4.1 Qualitative evaluation

The Stanford Bunny is a well known dataset used for comparison of reduction methods, see Figure 6.5. The dataset itself is very small compared to the target processing sets.

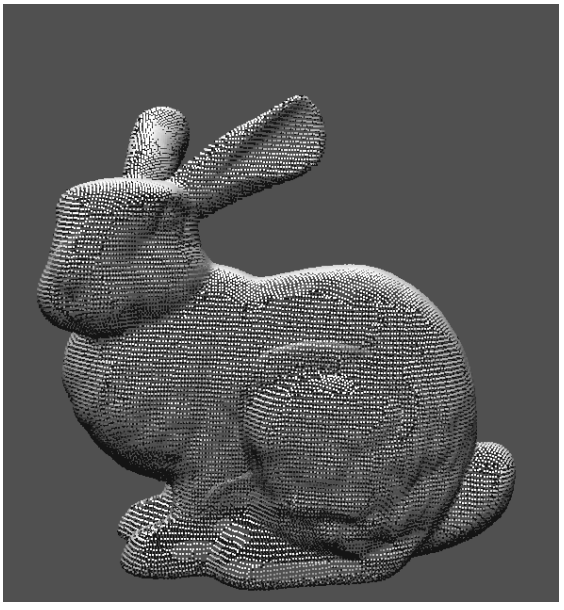


FIGURE 6.5: Stanford Bunny dataset

Figure 6.6 shows the visual comparison of two decimation methods, every n^{th} decimation and our volumetric decimation, for a high volume reduction of the Stanford Bunny. Both results contain almost the same number of points. The every n^{th} element decimation contains noticeable clustering of points on the surface creating linear patterns, whereas the volumetric decimation retains more uniform point distribution.

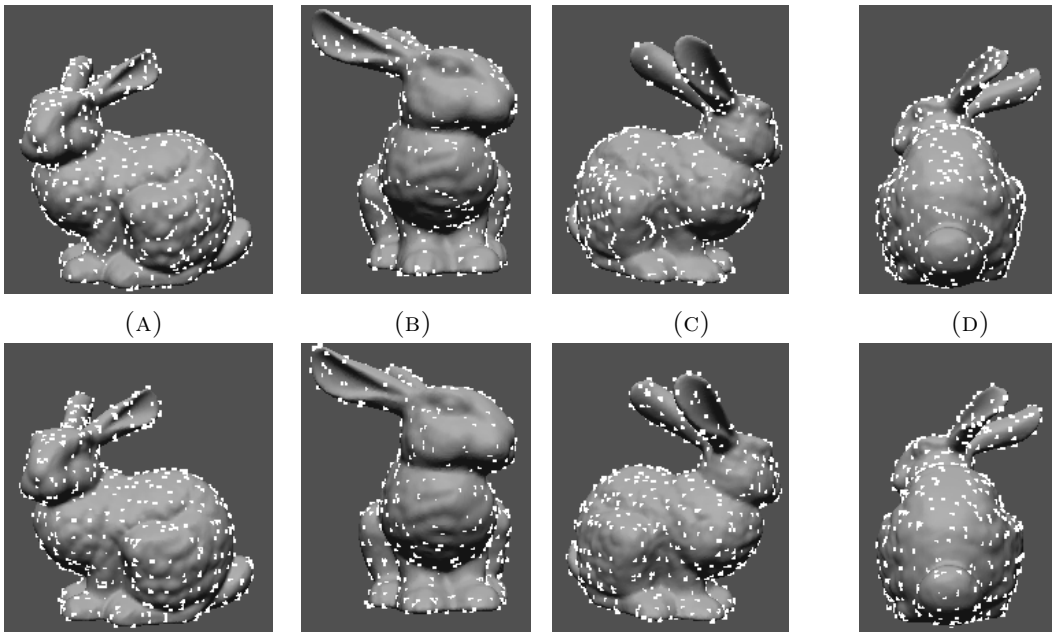


FIGURE 6.6: Visualisation of reduction results on Stanford Bunny dataset, the white dots represent the reduction results. $(A)-(D)$ show every n^{th} elimination and $(E)-(H)$ show reduction using our method

Figures 6.6 $(A)-(D)$ show every n^{th} elimination and $(E)-(H)$ show our method. The white points are the points retained after decimation. In Figures 6.6 (A) and (E) we can observe that both methods seem to be equally good with (A) showing some points following curved lines on the surface of the bunny whereas (E) shows more uniform distribution. The other views show more pronounced differences with the every n^{th} elimination clearly grouping points on curves placed on the surface leaving large patches on the chest area, tail and the bunny's ear without any points. Our method manages to cover the whole bunny in a more uniform fashion. Both methods are data dependant, yet our method will guarantee the coverage of the whole object.

Multiple locations were processed during the experimentation including ironworks site and various chapels. One of them, Hen Gapel, is a small chapel in mid Wales, standing on a cross road, with several buildings in close vicinity, see Figure 6.7. This location is a good example of a rural location.

Figures 6.8 and 6.9 show the visual comparison of the two decimation methods for the Hen Gapel dataset. Both datasets contain comparable number of points. The every n^{th}

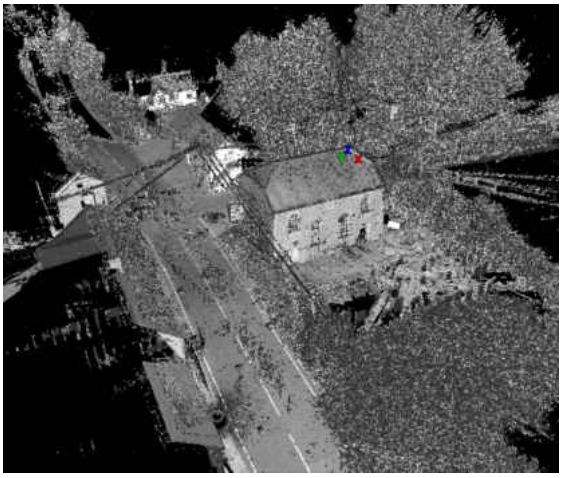


FIGURE 6.7: Hen Gapel dataset

element decimation loses point density towards the edges of a scene making the features unreadable.

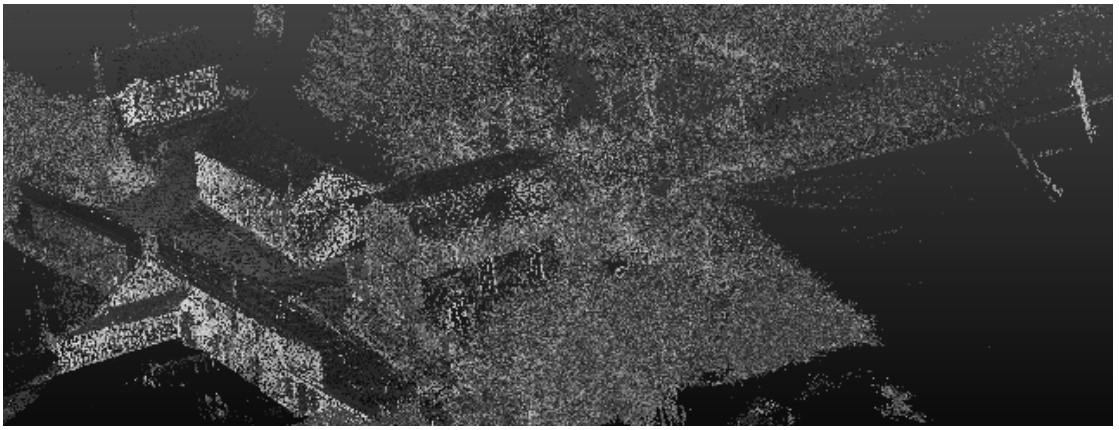


FIGURE 6.8: Volumetrically decimated (our method) Hen Gapel dataset: 2.5M points

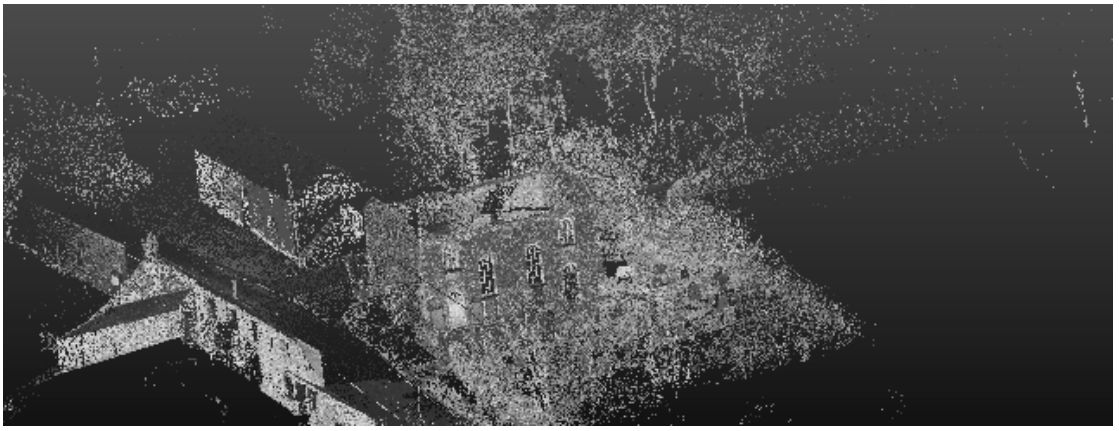


FIGURE 6.9: Every n^{th} element decimated Hen Gapel dataset: 2.5M points

Figures 6.10 and 6.11 show the chapel itself. The every n^{th} decimation depicts a superior quality when it comes to the facade of the chapel and its immediate surroundings, this is due to the fact that the chapel was the main target of the given laser scan and contains

data from many overlapping scans. When considering the area across the fencing wall the quality drops drastically with areas almost completely deprived of points. This quality inconsistency is one of the main drawbacks. Our method sacrifices the quality of the facade in favour of consistency. All the surrounding areas are presented in the same point density as the chapel itself. Our method allows the user to easily control point density of the output by sacrificing control over the total number of points.

FIGURE 6.10: Volumetrically decimated (our method) Hen Gapel dataset: chapel

FIGURE 6.11: Every n^{th} element decimated Hen Gapel dataset: chapel

Figures 6.12 and 6.13 perfectly showcase the inconsistency of quality in every n^{th} elimination method. Note the lack of data for parts of the road and the right hand side building. Our method shows the benefits of consistent point density by keeping all the details within the scene.

FIGURE 6.12: Volumetrically decimated (our method) Hen Gapel dataset: terraced cottages

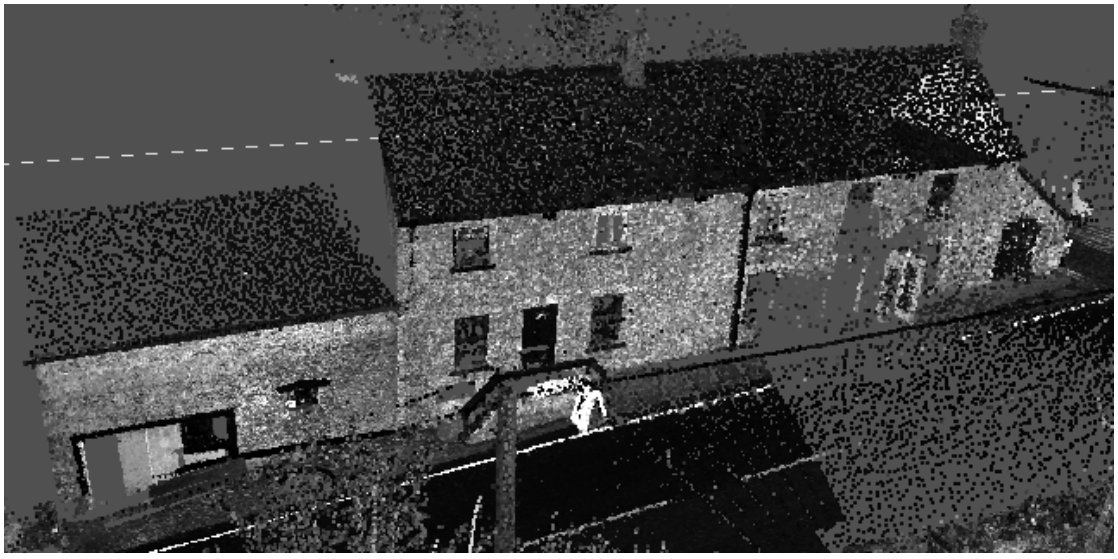


FIGURE 6.13: Every n^{th} element decimated Hen Gapel dataset: terraced cottages

Figures 6.14 and 6.15 show a small cottage at the very edge of our dataset. This cottage is clearly recognisable in the dataset decimated using our method, even though the point density is lower than the desired due to lack of data. The every n^{th} elimination method tends to struggle to capture the detail of the boundary of the dataset.

The Denbigh dataset represents an indoor location, namely St.Marcella's church in Denbigh. It is a complex structure with varied furnishings. Figures 6.16 and 6.17 show the visual comparison of the two decimation methods for the Denbigh dataset. Both datasets contain the same number of points. The every n^{th} element decimation loses point density towards the edges of a scene making the features unreadable. Note the detail of the pews, columns and the ladder leading to the bell tower. The noise visible by the cross

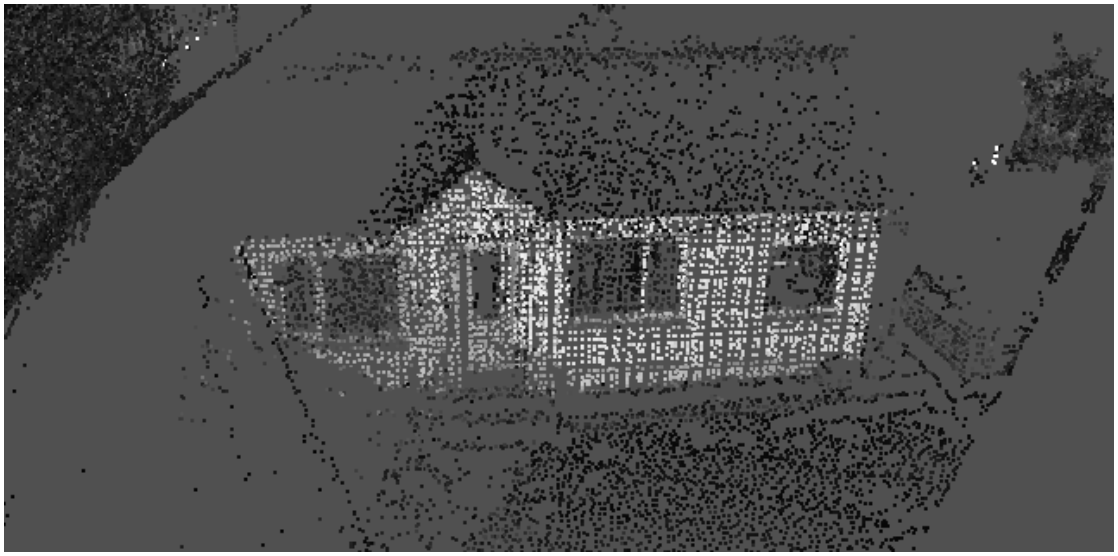


FIGURE 6.14: Volumetrically decimated (our method) Hen Gapel dataset: far cottage

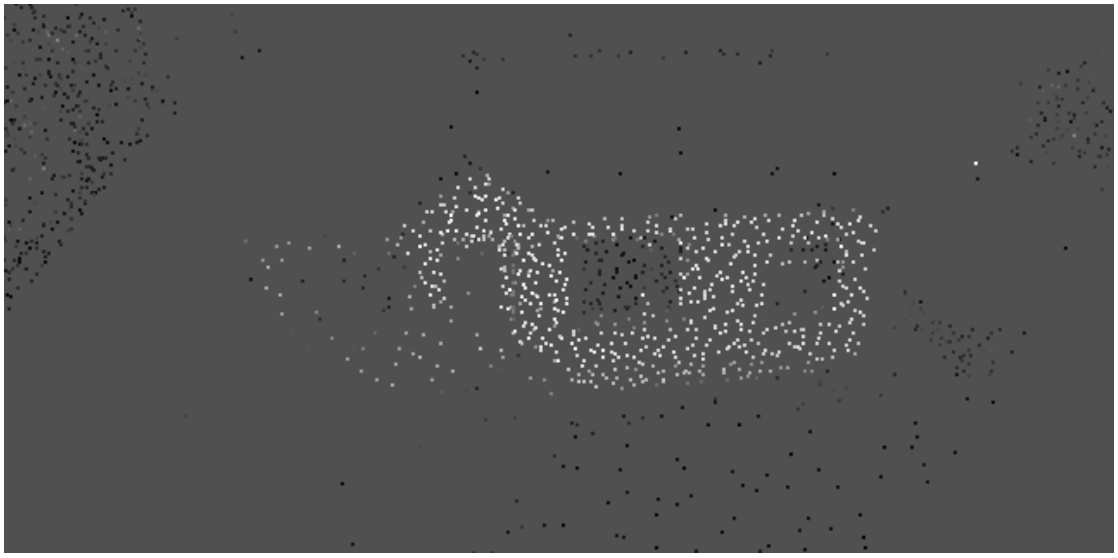


FIGURE 6.15: Every n^{th} element decimated Hen Gapel dataset: far cottage

showcases the drawback of our method which is potential retention of noise. In this case the noise it is caused by the imperfect match during the registration process.

Figures 6.18 and 6.19 showcase the display area by the entrance of the church. Note how the every n^{th} elimination method struggles to capture any data that was occluded in the high density scan, making occlusions visible. Our method manages to capture the floor, details of the plaque on the wall as well as the soft toys in the background showing little signs of occlusions.

Figures 6.20 and 6.21 show superior detail in the patches corresponding to the closest scan, yet lack detail in the areas occluded in the said scan, note the lack of data below the chairs on the right hand side and between the seat and the organ with every n^{th} method. Our method deals with those issues providing consistency throughout the dataset.

FIGURE 6.16: Volumetrically decimated (our method) Denbigh dataset: 33M points

FIGURE 6.17: Every n^{th} element decimated Denbigh dataset: 33M points

6.4.2 Quantitative evaluation

In this section our method is compared to the every n^{th} reduction method, the time comparison shows the viability of our method in comparison to the fastest available reduction method while replicating the result of volumetric reduction. The visual comparison acts only as a visual aid, as we do not aim to compare the viability of volumetric reduction to the every n^{th} element reduction.

The Stanford Bunny [100] dataset contains 35,947 points in a 2.5MB file. Table 6.1 shows the processing times for various methods. The performance of both methods is within the same order of magnitude in terms of time.

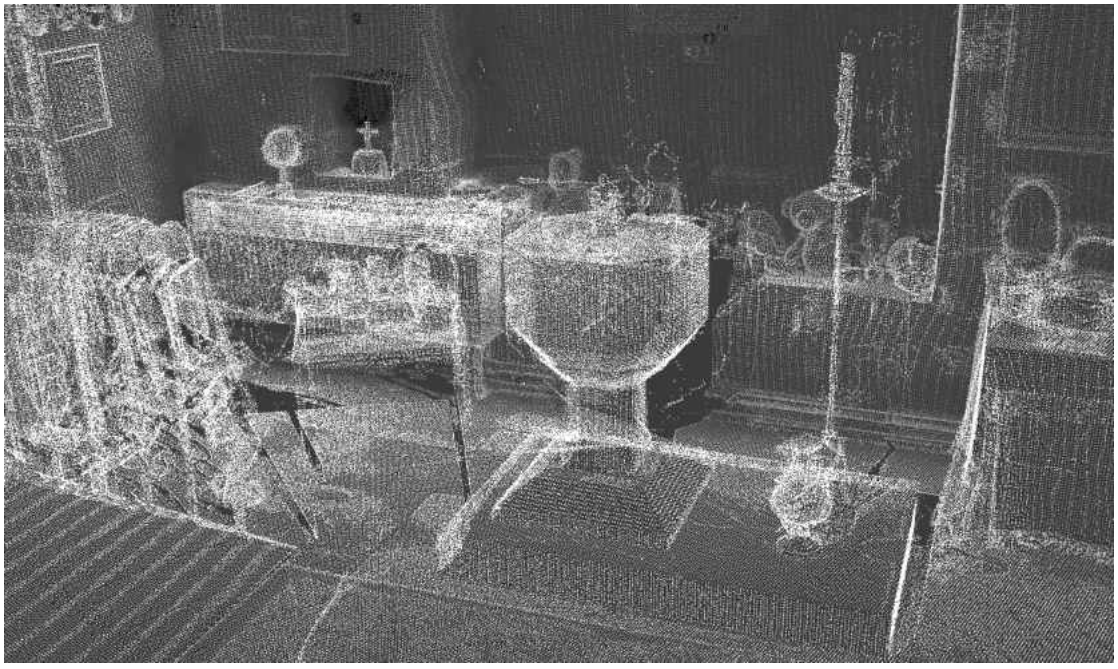


FIGURE 6.18: Volumetrically decimated (our method) Denbigh dataset: tables

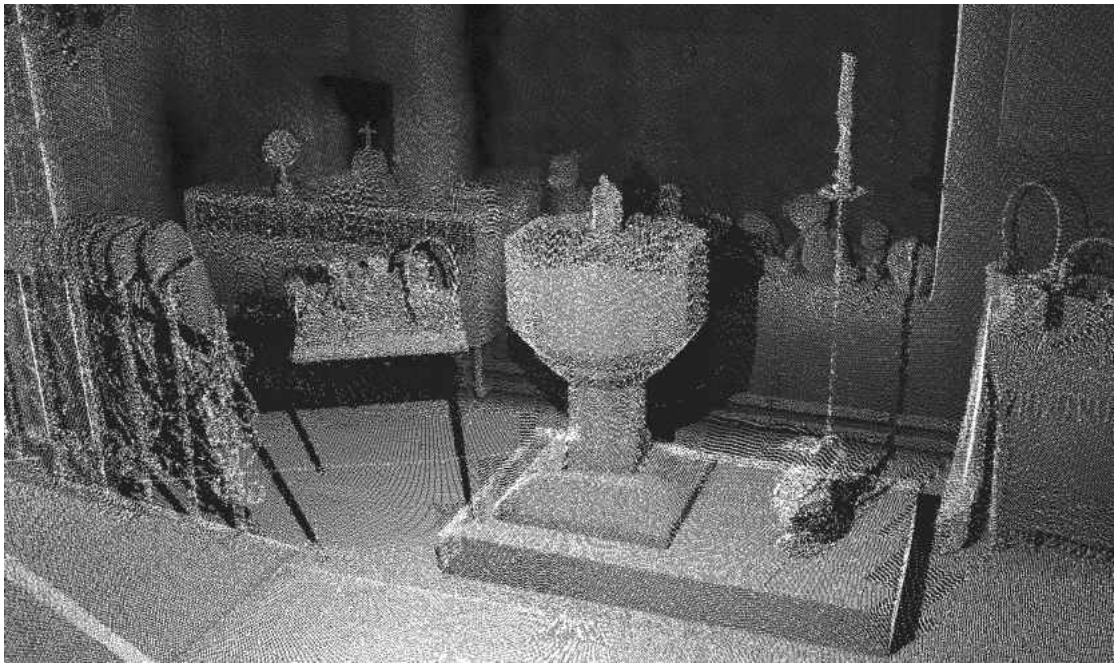


FIGURE 6.19: Every n^{th} element decimated Denbigh dataset: tables

The Hen Gapel dataset contains over 1,880 million points in a single 64GB file. Table 6.2 shows the processing times for various methods and decimation levels. The proposed method reduced the dataset with a time comparable to the every n^{th} element algorithm. Even though the time taken is almost double it still falls within the same order of magnitude. The processing time increases when the number of points in the reduced dataset increases.

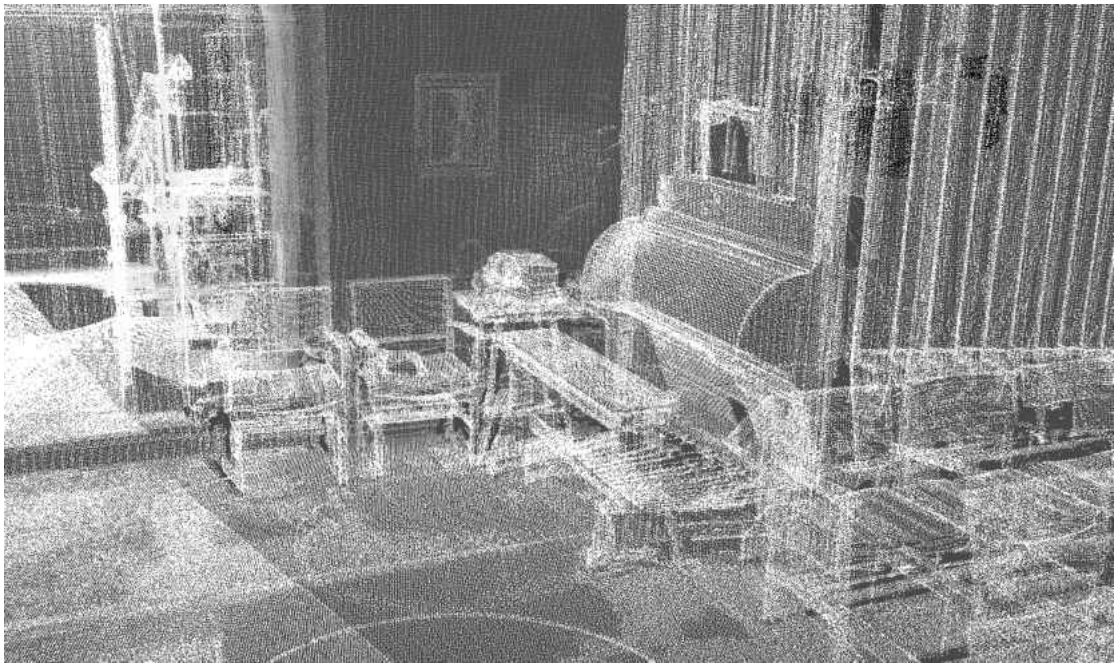


FIGURE 6.20: Volumetrically decimated (our method) Denbigh dataset : piano

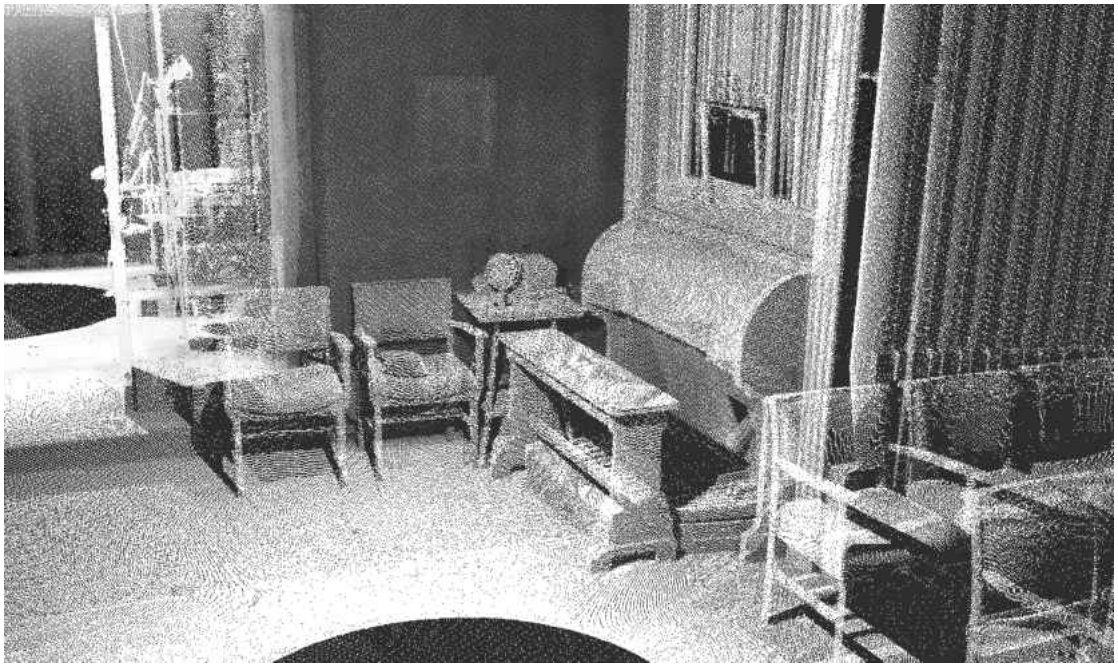


FIGURE 6.21: Every n^{th} element decimated Denbigh dataset : piano

The Denbigh dataset contains 1,857 million points in a single 55GB file. Table 6.3 shows the processing times for various methods. The proposed method performance is comparable with the every n^{th} element reduction.

The proposed method seems to be working with a time within the same order of magnitude as every n^{th} element decimation for the comparable point count in the resulting reduced point cloud. Both methods are capable of working with files that were too large

Table 6.1: Processing time for the Stanford Bunny dataset

Method	Result point count	Processing time
Line count	N/A	230ms
Every n^{th} point	765	420ms
Proposed method	761	553ms

Table 6.2: Processing time for the Hen Gapel dataset

Method	Result point count	Processing time
Export to XYZI from cyclone	1,880 million	17h
Line count	N/A	16min
Every 752nd point	2.5 million	27min
Proposed method	2.5 million	55min
Every n^{th} point	54 million	97min
Proposed method	54 million	145min

Table 6.3: Processing time for the Denbigh dataset

Method	Result point count	Processing time
Export to XYZI from cyclone	1,857 million	11h
Line count	N/A	19min
Every n^{th} point	33 million	82min
Proposed method	33 million	104min

to be tested with the current implementation of the volumetric decimation algorithm in the Point Cloud Library [96], a current state of the art point cloud processing library.

6.4.3 Different implementations and potential optimisations

The proposed method can be implemented in many ways. The choice of knowledge base implementation will have the largest impact on the performance of the method. Even though a simple list could be used, the recommended structures include a hash or a n-ary tree, see Section 6.3.4. Even though this is very intensive on read/write operations, therefore quite slow and unnecessarily increasing the wear of a hard drive an experimental version was capable of processing a 3,000 million point file within 27 days returning a 230 million points decimated point cloud with peak memory usage of 17MB, including the memory used by the perl interpreter. The low memory requirement and computational requirements mean it could be used on low power platforms, such as Raspberry Pi.

The hash function used in tests is a modified version of Equation 6.1. Given the properties of the resolution following the magnitudes of the decimal system the division was substituted by retention of a specific number of decimal places, further reducing the computational complexity.

There is a possibility of further optimisation of the system by the use of hexadecimal numbers to represent individual codes in the hash code. The suggested mapping is from digits and signs to hexadecimal values/glyphs. Such hexadecimal code would use a 16-ary tree to store the values. The main benefit of using the glyphs is a vast reduction of memory requirement to store the knowledge base. Alternatively, a hash map based representation can be used to further limit the memory requirements. We have tested a 16-ary tree representation and it provided the memory requirements of around 10 times the size of the resulting decimated dataset.

Due to the large quantity of data, the overhead and number of involved pointers play a major role in the memory requirements of the implementations.

6.4.4 Conclusions

Point clouds increase in size and become even harder to process. Decimation is a logical step to allow processing and viewing of the reduced resolution datasets. The presented method is capable of processing large datasets (upward of 1,900 million points) within a reasonable time frame, with processing time comparable to every n^{th} element elimination. The proposed method returns fixed point density clouds without altering the surface points. Our method was shared with project partners working on the 3D reconstruction of the gathered scenes. The feedback we acquired from the 3D modelling studios, that tested the method, suggest that our method is preferred over the every n^{th} elimination. Our method has shown to be more suited to the large data quantities than the conventional volumetric decimation as it does not require the creation of an octree. The processing time for high decimation values is within the order of magnitude of the every n^{th} elimination, making it competitive. The flexibility of the method at the implementation stage means that it can be tailored to the task and available resources.

In summary we have provided a novel method for replicating the results of the volumetric decimation while shifting the memory requirements to the result space rather than dataset space. The method can be used in applications that require comparison of two, or more, point clouds, such as map building, object detection and robotic navigation.

Chapter 7

Discussion and Conclusions

7.1 Introduction

This PhD is an integral part of the project investigating the process of laser scanning together with The Royal Commission on the Ancient and Historical Monuments of Wales. This chapter reviews the content of this thesis in relation to the project, provides a discussion about the viability of automation of the laser scanning process, summarises the answer to the research question, outlines some limitations of the presented methods and provides possible direction for future work.

7.2 Project conclusions

The project itself was a cooperation between academics and the Royal Commission on the Ancient and Historical Monuments of Wales. It involved data acquisition and processing as well as the exploration of improvement of the process. Some of the outcomes of the project resulted in scientifically significant outcomes, such as visibility measure, Chapter 5, and the decimation method, Chapter 6. Other outcomes involve the analysis of the environments laser scanned during the project to evaluate the scanning process and different strategies for scanner placement. This section will discuss the main objectives of the project and how they were met.

7.2.1 Automation

Our proposed automation of the laser scanning process requires a scene reconstruction method that is capable of creating a three dimensional metric map of the environment.

Most of the explored methods were struggling with the locations considered in this work, due to the lack of contrast in the hue in the scene. The outdoor locations were either grass covered earthworks or a mud filled river estuary. The indoor locations were mostly poorly lit areas with beige walls, dark brown floors and wooden furnishings. All in all, most of the environments were not properly extracted by the conventional reconstruction methods.

With no reliable way of environment reconstruction, the viability of automation is questionable. Additionally the proposed method is too computationally expensive to be used in the field. With improved methods and better reconstruction techniques the automation of parts of the scanning process will become viable. Unfortunately to automate the scanner placement we need robots capable of navigating environments that are often problematic for humans. This is not an unachievable goal, but the current technology does not provide robots capable of doing so in a non destructive manner, which is critical for the survey application. To summarise, the automation is possible providing further developments of the required methods, but it is not viable today.

It is possible to automate parts of the process. Further efforts should be directed toward the operator. A set of directives and simple tools that would help with the analysis of the environment and processing of the data would have a greater impact than a fully automated system.

7.2.2 Optimisation of the number of required scans

For a human operator reduction of the number of scans relies on the ability to visualise the result of the scan on location and ability to estimate the sufficient level of overlap between two scans. This knowledge is a result of the accumulated experience scanning on location and processing/registering the data. The variety of environments makes it hard to create specific guidelines and the generic guidelines are easily misunderstood. In Chapter 5 we have presented a method capable of detecting a set of laser scan positions that ensure the highest completeness within the set constraints. This method could be used as the basis of detection by an operator or used to verify and optimise an operator's own detected positions.

7.2.3 Laser scanning process

As described in the Chapter 2, the laser scanning process consists of four stages: reconnaissance, position detection, scanning and registration. Reconnaissance and position detection are highly dependant on the environment in question. In some environments,

such as the medieval bridge and Brymbo iron works, the accessibility is very limited due to debris after a collapse or natural formation such as a river bank. In those cases the positions are chosen incrementally by using the least approachable locations first and then attempting to fill in the gaps. Locations such as the tomb and Ynyslas Wreck provide vast areas and almost limitless amount of potential scanning position. However, in the case of the Ynyslas wreck the scanner position was limited by the river on one side and the riverbank on the other with mud making movement more difficult. The strategy for those is to position the scanner in a manner that captures the most interesting parts of the locations first. The last type of environment are complex indoor locations such as Grosmont, Denbigh and Bethania. Those pose a different issue. The specific architectural features have to be prioritised and the scanner positions have to take in account a 360 degree view.

The scanning of the outdoors locations is rather straightforward, the operator has to make sure that each of the faces of the object are visible in at least one scan, ensuring the overlap between multiple scans. The indoors environments pose more of a challenge as they provide multiple self occlusions. The operator has to consider the surroundings as separate smaller locations and then plan scans that cover those smaller locations making sure there is enough overlap between scans.

7.2.4 Usability of resulting large point clouds

The resulting point clouds contain from hundreds to thousands of millions of points which makes them difficult to handle, and often even load in the desired software package. To deal with that issue a set of tools are needed that are capable of extracting the desired information from the large point clouds. We have proposed a volumetric decimation method that is more memory efficient while matching the results of the conventional volumetric decimation. Such decimation methods allow the users to work with and visualise the point clouds.

7.3 Conclusions

This section revisits the original hypothesis and the research question specified in Section 1.3.

The hypothesis for this thesis was:

> "It is possible to eliminate the reliance on expert knowledge in terrestrial laser scanning by partially automating the process."

The research question for this thesis was:

> "Is a low resolution representation of a real-world 3D environment sufficient to approximate the completeness of a multi viewpoint visibility estimation of that environment?"

It has been shown that in the context of spatial reasoning, low resolution data can be used to reason about the higher resolution environments. The technique returns results comparable with the ones a human operator would provide. Even the lower resolution data provides sufficient information about the geometry of the scene. This information is then used to generate the desired positions of a laser scanner to ensure completeness of a scan. Even though full automation is not yet viable, replicating the results of an expert operator is possible. After the scanning stage, simple tools can help to manage the unwieldy datasets making them more user friendly. All in all expert knowledge cannot be replaced as the estimation of the quality of the captured data comes from understanding the process and spatial awareness. We are confident that it is possible to bridge the gap between a novice and expert operator and reduce the reliance on the expert knowledge.

7.3.1 Key contributions

The following are the key contributions developed during the project in the field of spatial reasoning as well as point cloud processing.

Useful developments were made to improve the environment perception and spatial reasoning, mainly related to data representation and understanding, see Chapter 4. This includes a new volumetric way of representing 3D data, that aids spatial reasoning. This hybrid implementation is designed to provide polygon like interaction with the data, while still retaining the volumetric nature of the dataset.

A new multi viewpoint visibility coverage estimation method was developed, see Chapter 5. The proposed visibility estimation system is capable of approximating the visibility coverage at a higher resolution, providing a slight underestimate of the visibility as the

resolution decreases. We have shown that our method performs similar to the human operator in complex indoor locations. The method is able to aid a novice operator as well as evaluate a set of viewpoints chosen by a novice operator.

The viewpoint position detection described in Chapter 5 is providing a set of viewpoints within the given environment that is of comparable quality to a set chosen by a human operator.

A point cloud reduction algorithm was developed, see Chapter 6, that allows a point by point fixed point density reduction in a volumetric manner, without the need for creating an octree based representation. It requires a fraction of the memory required by the conventional methods and only requires a single iteration of the dataset.

The viability of automation is discussed in Section 7.2.1. To reiterate, the automation of the laser scanning process, due to the issues related to acquisition of the initial dataset, traversability within the environment and requirement for the operator input and supervision, although possible, is not yet viable. New technologies for scene reconstruction have to become commercially available to aid the automation process.

7.4 Limitations

The hybrid data representation was specifically designed to work in the visibility context and is limited to the applications that require processing using line of sight methods. This limits the potential applications of the data representation to mainly rendering or spatial analysis.

The current implementation of the visibility estimation method uses inefficient detection of the set of potential viewpoints. Initially designed to be used in the field, the method is too computationally expensive. It could, however, be used as an evaluation of the positions chosen by the operator. This brings us to another drawback, the method relies on an accurate metric representation of the environment collected by the operator. As previously mentioned many existing techniques are not suitable for the types of the environments considered in the project.

The main limitation of the decimation method is the reliance on the implementation. Thanks to its flexibility, that allows tailoring the implementation to the users requirements, it becomes unnecessarily difficult to use. A general purpose version is required to mitigate this isssue.

7.5 Future work

The data representation presented in Chapter 4 will be implemented as a standalone package based on sparse octree, alongside an icoseptree (27-ary tree) implementation and the different implementation tested for raycasting and voxel culling. The renderer will be reimplemented to work with the optimised version of the tree.

The visibility estimation method presented in the Chapter 5 will be optimised to allow real time monitoring of the scenes. Various potential viewpoint set generation methods will be implemented and exhaustively tested to see if the method could be run in real time. The system can be extended to generate occlusion volumes that show exact volume of occluded space instead of just showing the currently occluded surfaces. This could be used to highlight the areas of interest to supplement the viewpoint detection to cover the whole volume.

Another desired improvement is the implementation of the area of interest, where the operator would designate parts of the scene as more important. The system would then provide the best coverage while concentrating solely on the coverage of the area of interest.

The decimation method presented in Chapter 6 will be implemented as a plugin for the existing point cloud processing toolkits as well as common 3D modelling software to allow others to load decimated point clouds. A mutli-scale loader will be created to allow viewing of the full resolution dataset while zoomed in on the surface of the object.

7.6 Alternative Applications

The visibility estimation method was designed to tackle the muliti-viewpoint visibility problem. As such it satisfies the requirements of illuminance satisfaction problem, which premise is illumination of the area using the least amount of light fixtures. This means the method can be used to design the illumination for any given scene.

Another potential application is in security situation where an existing scene can be monitored in real time for occlusions that could be exploited by thieves or suggest that the monitored object is not in sight.

The last potential application is in training of laser scanner operators, where novice users can have their placement of the laser scanner evaluated by the system without the need to do the actual scans.

7.7 Publications and attended Conferences

The following conferences were attended during the project to present the relevant findings:

- International Conference on Computer Vision Theory and Applications (VISAPP) 2014, Lisbon, Portugal
- RIVIC Graduate School 2013, Bangor, UK
- Digital Past 2013: New technologies in heritage, interpretation and outreach, Monmouth, UK
- Robotics innovation for Cultural Heritage (RICH) 2012, Venice, Italy
- Digital Past 2012: Digital Technologies and Heritage, Lllandrindod Wells, UK

One conference publication was made during the project:

Marek Ososinski, Frédéric Labrosse. *Multi-viewpoint Visibility Coverage Estimation for 3D Environment Perception - Volumetric Representation as a Gateway to High Resolution Data.* VISAPP (2) 2014: 462-469.

There are two additional journal publications awaiting submission:

- Point by point volumetric decimation of large point clouds.
- Influence of the initial set generation on multi viewpoint visibility estimation.

Bibliography

[1] Raimund Karl, Jonathan Roberts, Andrew Wilson, Katharina Möller, Helen C. Miles, Ben Edwards, Bernard Tiddeman, Frédéric Labrosse, and Emily La Trobe-Bateman. Picture this! community-led production of alternative views of the heritage of gwynedd. *Journal of Community Archaeology & Heritage*, 1(1):23–36, 2014. doi: 10.1179/2051819613Z.0000000003.

[2] Luz A. Torres-Méndez and Roberto Cervantes-Jacobo. Learning cognitive human navigation behaviors for indoor mobile robot navigation. In *COGNITIVE 2012, The Fourth International Conference on Advanced Cognitive Technologies and Applications*, pages 37–45, July 2012.

[3] Dimitris Papadias and Timos Sellis. Qualitative representation of spatial knowledge in two-dimensional space. *The VLDB Journal*, 3(4):479–516, October 1994. ISSN 1066-8888.

[4] J. M. Bindewald, M. E. Miller, and G. L. Peterson. *A function-to-task process model for adaptive automation system design.* International Journal of Human-Computer Studies, Volume 72, Issue 12, December 2014, 2014.

[5] David Jones. *3D Laser Scanning for Heritage.* English Heritage, Swindon England, 2011. ISBN product code 51704.

[6] Y. Engestrom, R. Engestrom, and M. Karkkainen. *Polycontextuality and boundary crossing in expert cognition: learning and problem solving in complex work activities.* Learn Instr, 5 (1995), 1995.

[7] K. A. Ericsson, Krampe R. Th, and C.Tesch-Romer. *The role of deliberate practice in the acquisition of expert performance.* Psychol Rev, 100 (1993), 1993.

[8] K. A. Ericsson, N. Charness, and P. J. Feltovic. *The influence of experience and deliberate practice on the development of superior expert performance.* The Cambridge handbook of expertise and expert performance, Cambridge University, Cambridge (2006), 2006.

[9] D. Z. Hambrick, F. L. Oswald, E. M. Altmann, E. J. Meinz, F. Gobet, and G. Campitelli. *Deliberate practice: is that all it takes to become an expert?* Intell, 45 (2014), 2014.

[10] B. N. Macnamara, D. Z. Hambrick, and F. L. Oswald. *Deliberate practice and performance in music, games, sports, education, and professions: a meta-analysis.* Psychol Sci, 25 (2014), 2014.

[11] Mitchell Booth. *Getting started with ArcGIS.* ESRI, Redlands, CA, 1999.

[12] V. Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory Series B*, 18:39–41, 1975.

[13] Steve Fisk. A short proof of chvátal's watchman theorem. *Journal of Combinatorial Theory, Series B*, 24(3):374, 1978. ISSN 0095-8956. doi: http://dx.doi.org/10.1016/0095-8956(78)90059-X.

[14] Joseph Rourke. *Art gallery theorems and algorithms.* Oxford University Press, New York, 1987. ISBN 0-19-503965-3.

[15] Brad Ballinger, Nadia Benbernou, Prosenjit Bose, Mirela Damian, Erik D. Demaine, Vida Dujmović, Robin Flatland, Ferran Hurtado, John Iacono, Anna Lubiw, Pat Morin, Vera Sacristán, Diane Souvaine, and Ryuhei Uehara. Coverage with k-transmitters in the presence of obstacles. *Journal of Combinatorial Optimization*, 25(2):208–233, 2013. ISSN 1382-6905. doi: 10.1007/s10878-012-9475-x.

[16] Uğur Murat Erdem and Stan Sclaroff. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Computer Vision and Image Understanding*, 103(3):156 – 169, 2006. ISSN 1077-3142. doi: http://dx.doi.org/10.1016/j.cviu.2006.06.005. Special issue on Omnidirectional Vision and Camera Networks.

[17] Justin Iwerks and Joseph S.B. Mitchell. The art gallery theorem for simple polygons in terms of the number of reflex and convex vertices. *Information Processing Letters*, 112(20):778–782, 2012. ISSN 0020-0190. doi: http://dx.doi.org/10.1016/j.ipl.2012.07.005.

[18] James King and David Kirkpatrick. Improved approximation for guarding simple galleries from the perimeter. *Discrete & Computational Geometry*, 46(2):252–269, 2011. ISSN 0179-5376. doi: 10.1007/s00454-011-9352-x.

[19] Sarah Cannon, Diane L. Souvaine, and Andrew Winslow. Hidden mobile guards in simple polygons. *CoRR*, abs/1206.1803, 2012.

[20] F. Hoffmann, M. Kaufmann, and K. Kriegel. The art gallery theorem for polygons with holes. In *32nd Annual Symposium on Foundations of Computer Science*, pages 39–48, 1991. doi: 10.1109/SFCS.1991.185346.

[21] Giovanni Viglietta. Guarding and searching polyhedra; doctoral thesis; university of pisa. 2012.

[22] Javier Cano, Csaba D. Tóth, and Jorge Urrutia. A tight bound for point guards in piecewise convex art galleries. *International Journal of Computational Geometry*, 46(8):945–958, 2013. ISSN 0925-7721. doi: http://dx.doi.org/10.1016/j.comgeo.2013.04.004.

[23] Andrea Bottino and Aldo Laurentini. A nearly optimal algorithm for covering the interior of an art gallery. *Pattern Recognition*, 44(5):1048–1056, 2011. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/j.patcog.2010.11.010.

[24] Yoav Amit, Joseph S. B. Mitchell, and Eli Packer. Locating guards for visibility coverage of polygons. *International Journal of Computational Geometry*, 20(5): 601–630, 2010.

[25] Karl J. Obermeyer, Anurag Ganguli, and Francesco Bullo. Multi-agent deployment for visibility coverage in polygonal environments with holes. *International Journal of Robust and Nonlinear Control*, 21(12):1467–1492, Aug 2011. doi: 10.1002/rnc.1700.

[26] Bengt J. Nilsson. Guarding art galleries — methods for mobile guards; doctoral thesis; lund university. 1995.

[27] A. Bottino and A. Laurentini. A practical iterative algorithm for sensor positioning. In *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, volume 1, pages 4 pp.–1092, Sept 2005. doi: 10.1109/ETFA.2005.1612650.

[28] Görkem Safak. The art-gallery problem: A survey and an extension; masters thesis; royal institute of technology, kth csc. 2009.

[29] Franco Preparata. *Computational geometry : an introduction*. Springer-Verlag, New York, 1985. ISBN 3540961313.

[30] E. Trucco, M. Umasuthan, A.M. Wallace, and V. Roberto. Model-based planning of optimal sensor placements for inspection. *IEEE Transactions on Robotics and Automation*, 13(2):182–194, 1997. ISSN 1042-296X. doi: 10.1109/70.563641.

[31] M. Marengoni, B.A. Draper, A. Hanson, and R. Sitaraman. A system to place observers on a polyhedral terrain in polynomial time. *Image and Vision Computing*, 18(10):773 – 780, 2000. ISSN 0262-8856. doi: http://dx.doi.org/10.1016/S0262-8856(99)00045-1.

[32] O. Viro. *Elementary topology : problem textbook*. American Mathematical Society, Providence, R.I, 2008. ISBN 978-0-8218-4506-6.

[33] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[34] Wesley H. Huang and Kristopher R. Beevers. Topological mapping with sensing-limited robots. In Michael Erdmann, Mark Overmars, David Hsu, and Frank van der Stappen, editors, *Algorithmic Foundations of Robotics VI*, volume 17 of *Springer Tracts in Advanced Robotics*, pages 235–250. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-25728-8. doi: 10.1007/10991541_17.

[35] M. Neal and F. Labrosse. Rotation-invariant appearance based maps for robot navigation using an artificial immune network algorithm. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pages 863–870 Vol.1, June 2004. doi: 10.1109/CEC.2004.1330951.

[36] Wesley H. Huang and Kristopher R. Beevers. Topological map merging. In Rachid Alami, Raja Chatila, and Hajime Asama, editors, *Distributed Autonomous Robotic Systems 6*, pages 97–106. Springer Japan, 2007. ISBN 978-4-431-35869-5. doi: 10.1007/978-4-431-35873-2_10.

[37] W. A. Seymour. *A History of the Ordnance Survey*. Dawson, Folkestone, Kent, England, 1980. ISBN 0712909796.

[38] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31 (1-3):29–53, 1998. ISSN 0885-6125. doi: 10.1023/A:1007436523611.

[39] Joan Solà, Teresa Vidal-Calleja, Javier Civera, and JoséMaríaMartínez Montiel. Impact of landmark parametrization on monocular ekf-slam with points and lines. *International Journal of Computer Vision*, 97(3):339–368, 2012. ISSN 0920-5691. doi: 10.1007/s11263-011-0492-5.

[40] J. Civera, A.J. Davison, and J. Montiel. Inverse depth parametrization for monocular slam. *Robotics, IEEE Transactions on*, 24(5):932–945, Oct 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2003276.

[41] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *Robotics and Automation, IEEE Transactions on*, 17(3):242–257, Jun 2001. ISSN 1042-296X. doi: 10.1109/70.938382.

[42] J. Nieto, T. Bailey, and E. Nebot. *Scan-SLAM: combining EKF-SLAM and scan correlation, International Conference on Field and Service Robotics*. 2005.

[43] Wei Liu, Tao Wang, and Yachong Zhang. A relative map approach for efficient ekf-slam. In *Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese*, pages 2646–2650, Aug 2014. doi: 10.1109/CGNCC.2014.7007586.

[44] Regan David, Kenneth Beverley, and Max Cynader. The visual perception of motion in depth. *Scientific American*, 241(1):136–151, July 1979. doi: 10.1038/scientificamerican0779-136.

[45] A. Rituerto, L. Puig, and J.J. Guerrero. Visual slam with an omnidirectional camera. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 348 –351, aug. 2010. doi: 10.1109/ICPR.2010.94.

[46] Valérie Cornilleau-Pérès, Mark Wexler, Jacques Droulez, Emmanuel Marin, Christian Miège, and Bernard Bourdoncle. Visual perception of planar orientation: dominance of static depth cues over motion cues. *Vision Research*, 42(11):1403 – 1412, 2002. ISSN 0042-6989. doi: 10.1016/S0042-6989(01)00298-X.

[47] B. Williams and I. Reid. On combining visual slam and visual odometry. In *Proc. International Conference on Robotics and Automation*, 2010.

[48] Heng Wang, Shoudong Huang, Udo Frese, and Gamini Dissanayake. The nonlinearity structure of point feature SLAM problems with spherical covariance matrices, Automatica. 49:3112–3119, October 2013.

[49] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052 –1067, june 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1049.

[50] Cyril Roussillon, Aurélien Gonzalez, Joan Solà, Jean-Marie Codol, Nicolas Mansard, Simon Lacroix, and Michel Devy. Rt-slam: A generic and real-time visual slam implementation. In JamesL. Crowley, BruceA. Draper, and Monique Thonnat, editors, *Computer Vision Systems*, volume 6962 of *Lecture Notes in Computer Science*, pages 31–40. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23967-0. doi: 10.1007/978-3-642-23968-7_4.

[51] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *In Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2002.

[52] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34 –46, feb. 2007. ISSN 1552-3098. doi: 10.1109/TRO.2006.889486.

[53] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2443–2448, 2005.

[54] Takahiko Nakamura and Satoshi Suzuki. Simplified ekf-slam by combining laser range sensor with retro reflective markers for use in kindergarten. *International Journal of Robotics and Mechatronics*, 1(1), 2014. ISSN 2288-5889. URL `http://ojs.unsysdigital.com/index.php/ijrm/article/view/122`.

[55] Yao-Chang Chen, Tsung-Han Lin, and Ta-Ming Shih. Using range and bearing observation in stereo-based ekf slam. In Ashutosh Natraj, Stephen Cameron, Chris Melhuish, and Mark Witkowski, editors, *Towards Autonomous Robotic Systems*, volume 8069 of *Lecture Notes in Computer Science*, pages 340–352. Springer Berlin Heidelberg, 2014. ISBN 978-3-662-43644-8. doi: 10.1007/978-3-662-43645-5_37.

[56] Kin Leong Ho and Paul Newman. Loop closure detection in SLAM by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9):740 – 749, 2006. ISSN 0921-8890. doi: 10.1016/j.robot.2006.04.016. (ECMR '05) 2nd European Conference on Mobile Robots.

[57] U. Frese and L. Schroder. Closing a million-landmarks loop. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5032 –5039, oct. 2006. doi: 10.1109/IROS.2006.282531.

[58] A.J. Davison, Y. González Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon*, July 2004.

[59] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. International Conference on Computer Vision, Nice*, October 2003.

[60] Zhengyou Zhang. Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2): 4–10, Feb 2012. ISSN 1070-986X. doi: 10.1109/MMUL.2012.24.

[61] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the rgb-d slam system. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1691–1696, May 2012. doi: 10.1109/ICRA.2012.6225199.

[62] Khyati K.Vyas Deval Keralia and Khushali Deulka. Google project tango – a convenient 3d modeling device. *International Journal of Current Engineering and Technology*, 4(5):3139–3142, Oct 2014.

[63] Andreas Nuchter. *3D Robotic Mapping the simultaneous localization and mapping problem with six degrees of freedom.* Springer-Verlag, Berlin, 2009. ISBN 978-3540898832.

[64] Liang Zhao, Shoudong Huang, Lei Yan, and G. Dissanayake. Parallax angle parametrization for monocular slam. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3117–3124, May 2011. doi: 10.1109/ICRA.2011.5979934.

[65] Berthold K. P. Horn. Shape from shading. chapter Obtaining Shape from Shading Information, pages 123–171. MIT Press, Cambridge, MA, USA, 1989. ISBN 0-262-08183-0.

[66] A.H. Ahmed and A.A. Farag. Shape from shading under various imaging conditions. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383286.

[67] Rui Huang and W.A.P. Smith. Shape-from-shading under complex natural illumination. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 13–16, Sept 2011. doi: 10.1109/ICIP.2011.6115701.

[68] F. Ferrari, E. Grosso, G. Sandini, and M. Magrassi. A stereo vision system for real time obstacle avoidance in unknown environment. In *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, pages 703–708 vol.2, Jul 1990. doi: 10.1109/IROS.1990.262486.

[69] Xinjian Fan, Xuelin Wang, and Yongfei Xiao. A shape-based stereo matching algorithm for binocular vision. In *Security, Pattern Analysis, and Cybernetics (SPAC), 2014 International Conference on*, pages 70–74, Oct 2014. doi: 10.1109/SPAC.2014.6982659.

[70] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8): 1362–1376, 2010.

[71] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[72] M. Jancosek and T. Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3121–3128, June 2011. doi: 10.1109/CVPR.2011.5995693.

[73] S.K. Nayar and Y. Nakagawa. Shape from focus. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(8):824–831, Aug 1994. ISSN 0162-8828. doi: 10.1109/34.308479.

[74] M. Lenz, D. Ferstl, M. Ruther, and H. Bischof. Depth coded shape from focus. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–8, April 2012. doi: 10.1109/ICCPhot.2012.6215218.

[75] S.K. Nayar. Shape from focus system. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 302–308, Jun 1992. doi: 10.1109/CVPR.1992.223259.

[76] J.-M. Lavest, G. Rives, and M. Dhome. Three-dimensional reconstruction by zooming. *Robotics and Automation, IEEE Transactions on*, 9(2):196–207, Apr 1993. ISSN 1042-296X. doi: 10.1109/70.238283.

[77] Mojtaba Nouri Bygi, Shervin Daneshpajouh, Sharareh Alipour, and Mohammad Ghodsi. Weak visibility counting in simple polygons. *Journal of Computational and Applied Mathematics*, 288(0):215 – 222, 2015. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/j.cam.2015.04.018.

[78] Roger Villemaire and Sylvain Hallé. Reasoning about visibility. *Journal of Applied Logic*, 10(2):163 – 178, 2012. ISSN 1570-8683. doi: http://dx.doi.org/10.1016/j.jal.2012.02.001.

[79] Rouven Strauss, Florin Isvoranu, and Gershon Elber. Geometric multi-covering. *Computers & Graphics*, 38(0):222 – 229, 2014. ISSN 0097-8493. doi: http://dx.doi.org/10.1016/j.cag.2013.10.018.

[80] David J. Chalmers, Robert M. French, and Douglas R. Hofstadter. High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *J. Exp. Theor. Artif. Intell.*, 4(3):185–211, August 1992. ISSN 0952-813X. doi: 10.1080/09528139208953747.

[81] William Thompson. *Visual perception from a computer graphics perspective*. CRC Press, Boca Raton, Fla, 2011. ISBN 978-1568814650.

[82] Anna Berti and Giacomo Rizzolatti. Visual processing without awareness: Evidence from unilateral neglect. *J. Cognitive Neuroscience*, 4(4):345–351, October 1992. ISSN 0898-929X. doi: 10.1162/jocn.1992.4.4.345.

[83] N. Beschin, G. Cocchini, S. Della Sala, and RH. Logie. What the eyes perceive, the brain ignores: A case of pure unilateral representational neglect. *Cortex*, pages 3–26, March 1997. PubMed PMID: 9088719.

[84] Russell A. Brown. Building a balanced k-d tree in $o(kn \log n)$ time. *Journal of Computer Graphics Techniques (JCGT)*, 4(1):50–68, March 2015. ISSN 2331-7418.

[85] Yannis Manolopoulos. *R-trees theory and applications*. Springer, London, 2006. ISBN 1-85233-977-2.

[86] Wentai Liu and Ralph K. Cavin III. Rasterization theory, architectures, and implementations for a class of two-dimensional problems. *Integration, the VLSI Journal*, 6(2):179 – 199, 1988. ISSN 0167-9260. doi: http://dx.doi.org/10.1016/0167-9260(88)90038-7.

[87] Sascha Gundlach and Michelle K. Martin. *Mastering CryENGINE*. Packt Publishing - ebooks Account, 2014. ISBN 1783550252.

[88] Andrew Finch. *The Unreal Game Engine: A Comprehensive Guide to Creating Playable Levels*. 3DTotal Publishing, 2014. ISBN 1909414042.

[89] Marek Ososinski and Frédéric Labrosse. Multi-viewpoint visibility coverage estimation for 3d environment perception - volumetric representation as a gateway to high resolution data. In *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications, Volume 2, Lisbon, Portugal, 5-8 January, 2014*, pages 462–469, 2014. doi: 10.5220/0004693504620469.

[90] Paul Bryan. *Metric survey specifications for cultural heritage*. English Heritage, Swindon England, 2009. ISBN 978-1-84802-038-2.

[91] John Amanatides and Andrew Woo. A fast voxel traversal algorithm for ray tracing. In *Eurographics '87*, pages 3–10, 1987.

[92] D. D. Andrews. *Measured and drawn : techniques and practice for the metric survey of historic buildings*. English Heritage, Swindon, 2009. ISBN 9781848020474.

[93] Dariush Derakhshani. *Introducing Autodesk Maya 2015: Autodesk Official Press*. Sybex, 2014. ISBN 1118862848.

[94] Scott Onstott. *AutoCAD 2016 and AutoCAD LT 2016 Essentials: Autodesk Official Press*. Sybex, 2015. ISBN 1119059186.

[95] Michael Law and Amy Collins. *Getting to Know ArcGIS*. Esri Press, 2015. ISBN 1589483820.

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

[97] Hao Song and Hsi-Yung Feng. A global clustering approach to point cloud simplification with a specified data reduction ratio. *Computer-Aided Design*, 40(3):281 – 292, 2008. ISSN 0010-4485. doi: http://dx.doi.org/10.1016/j.cad.2007.10.013.

[98] Mark Pauly, Markus Gross, and Leif P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Proceedings of the Conference on Visualization '02*, VIS '02, pages 163–170, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7803-7498-3.

[99] Carsten Moenning and Neil A. Dodgson. A new point cloud simplification algorithm. In *In Proceedings 3rd IASTED Conference on Visualization, Imaging and Image Processing*, pages 1027–1033, 2003.

[100] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 311–318, New York, NY, USA, 1994. ACM. ISBN 0-89791-667-0. doi: 10.1145/192161.192241.