

Electronic Thesis and Dissertation Repository

10-26-2016 12:00 AM

Automatic Fracture Orientation Extraction from SfM Point Clouds

Jon Kissi

The University of Western Ontario

Supervisor

Dr K McIssac

The University of Western Ontario Joint Supervisor

Dr X Wang

The University of Western Ontario

Graduate Program in Electrical and Computer Engineering

A thesis submitted in partial fulfillment of the requirements for the degree in Master of
Engineering Science

© Jon Kissi 2016

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Computational Engineering Commons](#), [Electrical and Computer Engineering Commons](#),
and the [Software Engineering Commons](#)

Recommended Citation

Kissi, Jon, "Automatic Fracture Orientation Extraction from SfM Point Clouds" (2016). *Electronic Thesis and Dissertation Repository*. 4243.

<https://ir.lib.uwo.ca/etd/4243>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.

Abstract

Geology seeks to understand the history of the Earth and its surface processes through characterisation of surface formations and rock units. Chief among the geologists' tools are rock unit orientation measurements, such as Strike, Dip and Dip Direction. These allow an understanding of both surface and sub-structure on both the local and macro scale.

Although the way these techniques can be used to characterise geology are well understood, the need to collect these measurements by hand adds time and expense to the work of the geologist, precludes spontaneity in field work, and coverage is limited to where the geologist can physically reach.

In robotics and computer vision, multi-view geometry techniques such as Structure from Motion (SfM) allow reconstructions of objects and scenes using multiple camera views. SfM-based techniques provide advantages over Lidar-type techniques, in areas such as cost and flexibility of use in more varied environmental conditions, while sacrificing extreme levels of fidelity. Regardless of this, camera based techniques such as SfM, have developed to the point where accuracy is possible in the decimetre range.

Here is presented a system to automate the measurement of Strike, Dip and Dip Direction using multi-view geometry from video. Rather than deriving measurements using a method applied to the images, such as the Hough Transform, this method takes measurements directly from the software generated point cloud.

Point cloud noise is mitigated using a Mahalanobis distance implementation. Significant structure is characterised using a k-nearest neighbour region growing algorithm, and final surface orientations are quantified using the plane, and normal direction cosines.

Keywords: Structure from Motion, SfM, point cloud, Multi-View Geometry, Multi-View Stereo, Geology, Fracture Orientations, Strike, Dip Angle, Dip Direction, k nearest neighbor, region growing, Mahalanobis distance, direction cosine, computer vision, time series analysis, data mining

Dedicated to Esther, Dave, Benedicta and Jonathan.

If you treat an individual as if he were what he ought to be and could be, he will become what he ought to be and could be. - Johann Wolfgang von Goethe

There are only two mistakes one can make along the road to truth; not starting and not going all the way. - Buddha

Acknowledgements

Thank you to both my supervisors for the opportunity and patient help.

Thank you to my colleagues and contemporaries for their constantly helpful and encouraging words.

And thanks to the ECE ladies. Without your help navigating the wild rapids of *admin*, I would have drowned long ago. Thank you.



Contents

Abstract	ii
List of Figures	x
List of Tables	xii
1 Introduction	1
1.1 Motivation and Task Identification	1
1.2 Integrating Technology	2
1.2.1 The Introduction of Lidar	2
1.2.2 The Introduction of Multiple Views	3
1.2.3 Developments with Data Structures	4
1.3 The Monograph Focus	4
1.4 Executive General Summary	5
1.4.1 Initial Stages Incorporating Existing Solutions	6
1.4.2 Developing the Automatic System: GeoStrucure.m	6
1.5 Contributions of the Thesis	7
1.6 Thesis Overview	8
2 Literature Review	9
2.1 Geological Photogrammetry (Structure from 2D images)	9
2.1.1 Extracting Orientations: The Hough Transform	10
2.1.2 Extracting Orientations: The Fisher Constant and Distribution	10
2.2 Structure from Lidar and Light Ranging Technologies	11
2.2.1 Demonstrating the Total Station	11
2.2.2 Reconstructing on a Larger Scale	12
2.2.3 Automating Orientation Extraction	13
2.3 Structure from Motion	13

2.3.1	Scale Invariant Features	14
2.3.2	Automated Bundle Adjustment	15
2.3.3	SfM Pipeline Synthesis	16
2.3.4	SfM: Continuing Refinement	17
2.3.5	Densification	17
2.3.6	SfM: Utilising an End-to-End Framework	19
2.3.7	SfM: Developing Plug-n-Play Solutions	21
2.3.8	Integration of SfM and Geology	22
2.3.9	Data Acquisition Strategies: Exploring Camera Paths	22
2.4	Point Cloud Processing	23
2.4.1	Approaches to Segmentation	23
	Random Sample Consensus	23
	Tessellation and Meshing Techniques	24
	Techniques Incorporating Region Growing	24
2.4.2	Filtering Noise and Erroneous Points	25
2.5	Drone Platforms and Aerial Photogrammetry	26
2.5.1	Applications in the Field	26
	Strategies for Data Acquisition: Flight Plan	27
	Strategies for Data Acquisition: Number of Views	27
2.5.2	Integrated Methods in Geological Analysis	29
2.5.3	The Georeferencing Problem	29
3	Theoretical Background	31
3.1	Multi-View Geometry: Theoretical Underpinnings	31
3.2	SfM: SIFT	31
3.2.1	The Features of SIFT	32
	Scale Space Extrema Detection	32
	SSE Practical Application	33
	Keypoint Localisation	34
	Orientation Assignment	35
	The Local Image Descriptor	36
3.3	SfM: Bundle Adjustment and Reconstruction from Unorganised Datasets	37
3.3.1	Pre-Bundle Adjustment: Image Matching	37
	Image Matching Mechanics	38
3.3.2	Bundle Adjustment	39

	Bundle Adjustment: Theoretical Underpinnings	39
3.4	SfM: Patch-based Multi-View Steropsis	41
3.4.1	The Patch Model	42
3.4.2	Photometric Discrepancy Function	43
3.4.3	Image Model	43
3.4.4	Patch Reconstruction	44
	Initial Feature Matching	44
	Feature Matching	45
3.4.5	Patch Optimisation	45
3.4.6	Expansion	46
3.4.7	Filtering	46
3.5	Meshing/Tessellation: Poisson Surfacing	47
3.5.1	Surface Reconstruction: Combinatorial Techniques	47
3.5.2	The Poisson Technique	48
	Solving the Implicit Function: χ	48
3.5.3	Octrees	48
	Defining the Octree	49
3.6	Extracting Fracture Orientations	49
3.6.1	Dip Angle, Γ_α	50
3.6.2	Dip Direction, Γ_δ , and Strike, Γ_ζ	51
3.6.3	The Discontinuity Problem	52
4	Contributions	55
4.1	Overview Discussions	55
4.2	Stage 0: Data Acquisition	56
4.2.1	Flight Path	57
4.3	Stage 1: Preprocessing	58
4.3.1	Frame Separation	58
4.4	Stage 2: Initial Point Cloud generation (VSfM)	59
4.5	Stage 2: Point Cloud Densification (PMVS plugin)	59
4.6	Stage 2: Surface Tessellation (Meshlab)	59
4.6.1	Utilising PSR	60
4.7	Stage 2: GeoStructure.m	60
4.7.1	Mahalanobis Noise Mitigation	62
	Introducing Anisotropic Noise	62

	Understanding the Effectiveness of the Metric	63
4.8	Stage 2: The Voxel-fit Process (Matlab)	64
4.8.1	Creating the Point Space	64
4.8.2	Searching the Point Space	64
4.8.3	Applying the Voxel-fit	65
	Evaluating the Search Space	65
4.9	Stage 2: Segmentation through k NN Region Growing(Matlab)	67
4.9.1	Large Scale Structure through Region Detection	68
	Applying the Surface Threshold	68
	Applying the Planar Offset Threshold	69
4.9.2	Building the Region Plane	70
	Calculating the Plane Perimeter	71
	Minimum Perimeter Golden Section Search	72
	Applying the Rotation Matrix	73
4.10	Stage 3: Obtaining Fracture Orientations	75
5	Results and Analysis	77
5.1	General Rationale	77
5.2	Subject Choice Rationale	78
5.3	The Dataset	78
5.4	Result Set Considerations	79
5.5	Ground Truth	80
5.6	Obtaining Measurements	81
5.7	Fracture Orientation Comparison Quality Score	81
5.8	Measuring Pipeline Factor Efficacy	81
	Simulating Fracture Orientations with Rotation Matrices	82
	Measuring Differences Between Vectors	82
5.9	Experimental Results	84
5.9.1	Experiment 1: ζ	84
	Time Series	84
	Fracture Orientation Quality Score Analysis: Runs	85
	Fracture Orientation Quality Score Analysis: Regions	85
	Combined Interpretation	86
5.9.2	Experiment 2: θ	87
	Time Series	87

Fracture Orientation Quality Score Analysis: Runs	87
Fracture Orientation Quality Score Analysis: Regions	88
Combined Interpretation	88
5.9.3 Experiment 3: ψ	89
Time Series	89
Fracture Orientation Quality Score Analysis: Runs	89
Fracture Orientation Quality Score Analysis: Regions	90
Combined Interpretation	90
5.9.4 Experiment 4: k	92
Time Series	92
Fracture Orientation Quality Score Analysis: Runs	92
Fracture Orientation Quality Score Analysis: Regions	92
Combined Interpretation	93
5.9.5 Overall Evaluation	94
6 Conclusion and Future Work	95
6.1 Motivation	95
6.1.1 Mitigating the Software Generated Point Cloud Noise Problem	97
6.1.2 Development of the GeoStructure Algorithm Pipeline	97
6.2 Areas for Improvement and Future Work	97
Bibliography	99
Curriculum Vitae	105

List of Figures

1.1	Principle of Multi-View Geometry	3
1.2	Software Toolchain Workflow	5
2.1	Sparse Point Cloud	15
2.2	Densified Point Cloud	18
2.3	Commercially Available Drone	26
2.4	Camera Flight Plan Reconstruction	28
3.1	Difference of Gaussians (DoG)	33
3.2	Scale Space Extrema Detection	34
3.3	The Local Image Descriptor	36
3.4	Introduced Dense Cloud Noise	42
3.5	PMVS Surfel	43
3.6	Photometric Discrepancy Function	44
3.7	Pre-GeoStructure.m Processing	50
3.8	Fracture Orientation Geometry	51
4.1	Software Toolchain Workflow	56
4.2	Bundle Adjustment Camera Position Reconstruction	57
4.3	Introduced Densification Noise	58
4.4	PSR Tessellation Process	60
4.5	Software Toolchain Workflow	61
4.6	Mahalanobis Anisotropic Noise Filtering	63
4.7	Voxel Fitting Process	65
4.8	θ angular threshold	68
4.9	ψ Planar Offset	69
4.10	World/Region Plane Translation	71
4.11	Traversing the Region Plane	72
4.12	φ Golden Section Method	74

4.13	GeoStructure.m Processing	76
5.1	Subject and Instrument	77
5.2	Ground Truth Proceedure	78
5.3	Measurement Surfaces	80
5.4	Surface 6 Detail	81
5.5	ζ v time	84
5.6	ζ v $z(\mathbf{R})$	85
5.7	ζ v $\mathbf{R}(z)$	86
5.8	θ v time	87
5.9	θ v $z(\mathbf{R})$	88
5.10	θ v $\mathbf{R}(z)$	89
5.11	ψ v time	90
5.12	ψ v $z(\mathbf{R})$	91
5.13	ψ v $\mathbf{R}(z)$	91
5.14	k v time	92
5.15	k v $z(\mathbf{R})$	93
5.16	k v $\mathbf{R}(z)$	94

List of Tables

5.1	Table of Ground-truth Values	83
-----	--	----

Chapter 1

Introduction

This chapter of the report is divided as follows: First will be an outline and discussion on the motivation and aims for the thesis. Next will be an outline and discussion on the goals of thesis and how those goals will be achieved. Lastly, the contributions of the monograph will be highlighted, along with a brief explanation of how the contributions converge with the goals and aims of the thesis. Finally, there will be a brief overview of the layout of the rest of the document.

1.1 Motivation and Task Identification

A key aim of Geology is to understand the history of Earth and its surface processes, through characterisation of surface formations and rock units. In this regard, highly useful analytical quantities are fracture orientation measurements, such as *Strike*, *Dip Angle* and *Dip Direction*. Analysis using these identities allow structure to be inferred by characterisation of the relationship between [roughly] planar formations on the surface. These and other similar types of measurements help to formulate an understanding of the surface and sub-structure, across a range of scales.

Although the way this technique can be used to characterise geology is well understood, the need to collect these measurements manually challenges the geologist, adding time and expense, precluding spontaneity in field work, and, coverage is limited to where the geologist can physically reach. These are all very real limitations, that impact on the amount of data it is possible to gather on a target site, leading to the inevitable possibility of missing critical

information.

Despite increasing levels of technological integration, it is common for entire research degree projects to be focused on using fracture orientation techniques to characterise a single feature, with researchers spending weeks collecting measurements manually, and having data collection restricted by environmental conditions.

Fracture Orientation measurements are primary tools in the characterisation of surface features. The question is: can these processes be augmented to reduce time, resources and safety hazards, and, increase coverage in collection? Can the process of analysis and characterisation itself be automated, so that aspects of the collection, output and interpretation be improved?

1.2 Integrating Technology

As with many of other sciences, automation is having a dramatic impact on geology. There are various examples of research, incorporating aspects of data analysis, storage and visualisation technologies; use of robotics, computer vision and GIS. Focussing specifically on obtaining fracture orientations, investigation of the robotics and geology literature shows a specific interest, beginning in earnest between the mid 1990's to the early 2000's, in the extraction of structural information from remotely sensed datasets. That is, characterising areas of interest with standard orientation measurements derived from imagery, in a technique more generally referred to as *photogrammetry*. Various techniques, such as the Hough Transform, have been explored to take the fracture orientation as represented on the two-dimensional (2D) plane (the image) and infer or deduce the 3rd dimensional component [35, 46], thus characterising the structure under examination.

1.2.1 The Introduction of Lidar

After this period, several technological developments indicated new possibilities in this area. The most notable of these was the incorporation of Light Detection and Ranging (Lidar), and laser scanning-type technologies in general. While it would take a number of years for hardware development to make the technology practical in many areas, for the first time there was a technology that allowed for the direct characterisation of three-dimensional (3D) structure at range, without dealing with the health implications of Radar technology. This led to an increase

in the use of 3D point cloud structures in computer vision and image processing, engineering, along with development of techniques for processing and interpreting them.

1.2.2 The Introduction of Multiple Views

The second development was made possible by improvements in computing power and parallelisation, as well as improvements in practical size, resolving power of cameras, and Multi-View Geometry (MVG) techniques such as Multi-View Stereo (MVS) and Structure from Motion (SfM). Broadly speaking, these techniques allow the spatial reconstruction of objects/scenes from planar representations, using multiple views and angles. As is understood to be the case with stereo vision, the aggregation of views with differing angles implies *motion*, and the continuance of motion allows the extraction of *structure*.

In recent years, the integration of many computing, computer vision and robotics technologies has had a dramatic effect in Geology, and the introduction of drone (Unmanned Aerial Vehicles - UAV) technology as a survey tool, has opened up a great deal of opportunity to improve and diversify work in the area.

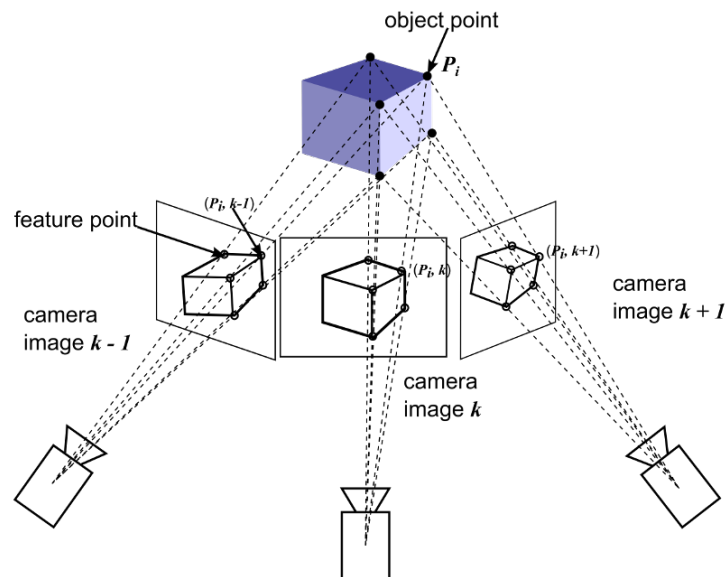


Figure 1.1: Visual representation of the principles of multi-view geometry. The capture of planar representations of an object from different locations allows the assignment of *feature points* to positions in 3D space. In turn, these allow the construction of a model approximating the original structure.

1.2.3 Developments with Data Structures

Recent work with point clouds has led to the development of several software packages [24, 63, 3] that processes point cloud datasets and extract fracture orientations from the point cloud representations. These encompass both developments using Lidar and SfM-type techniques as the data source. Many of these developments, especially on the SfM side, require user input to interpret the orientations, usually by indicating areas of interest on the original source images [59, 3, 56]. However, due to the precision and density of a Lidar point cloud, Gigli et al. [24] were able to extract the orientations from the point cloud.

The work by Gigli et al. is of particular interest because it is a fully automatic system, with no need for user intervention, taking “direct” measurements from the point cloud structure, rather than interpreting from 2D representations. A similar system using SfM, drone and camera technology would enable the geologist to perform accurate structure characterisation without deploying high cost, highly sensitive Lidar equipment in every situation.

1.3 The Monograph Focus

This forms the central focus of this work: to demonstrate a *Software Toolchain System* to extract fracture orientations directly from a point cloud representation, generated via SfM techniques. The aim is to do this in a *fully automatic* manner, that is, without any user interaction in the interpretation of measurements, or using the original images to infer the 3D structure, using a transform, or similar technique.

The implementation of the SfM algorithm is extremely complex but there are already various examples of applications that will perform more than well enough, for the purposes of this investigation. After the production of the initial point cloud from the SfM algorithm, several operations will have to be performed to get to a stage where meaningful geometrical quantities can be extracted. The proposal is to integrate several available software packages, along with bespoke algorithmic elements in to a Toolchain, that will receive as input unprocessed survey data, and output a processed spatial surface, with orientation measurements for each significant part of the surface.

These are the general stages that the proposed system must implement in order produce the desired output:

- Convert video to image frames,
- Process images to produce sparse point cloud,
- Process sparse point cloud to produce dense point cloud,
- Process dense point cloud to produce highly defined surface (as [24]),
- Process PSR to produce fitted surface,
- Develop algorithm extract fracture orientation measurements of major surfaces.

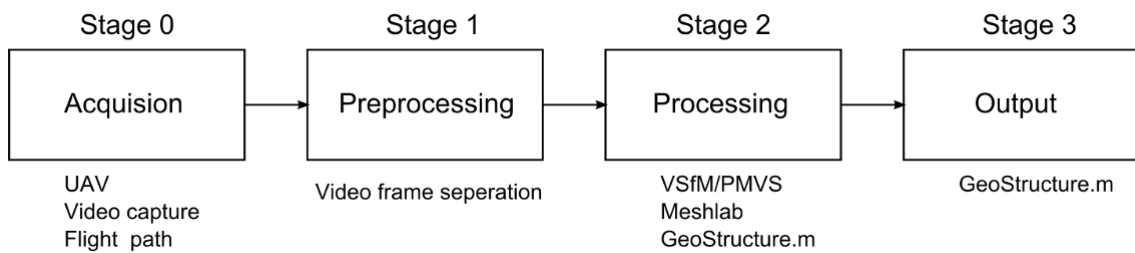


Figure 1.2: Overall work flow of the proposed system

1.4 Executive General Summary

Using the understanding and requirements stated above, it is possible to provide an executive summary of the major steps taken to develop a *fully automatic* system, for the extraction of Fracture Orientation measurements from a software generated point cloud.

Fig. 1.2 indicates that primarily the Toolchain is divided in half, where the first part of the processing is done by [mostly] existing software, in the areas of data acquisition, SfM point cloud production and surface tessellation (Stage 0, 1, 2). This section is primarily concerned with data collection and processing to obtain a smooth surface to operate on; attempting to replicate the highly defined surfaces of the Lidar-type point cloud [24]. The second part of the processing (Stage 2, 3), is done through GeoStructure.m, a series of Matlab programs, specifically developed to carry out the task of characterisation and extraction of Fracture Orientation measurements from the point cloud.

1.4.1 Initial Stages Incorporating Existing Solutions

Using a methodology based on the concept of “flight plan” in SfM, a regularised, repeating path of data collection was utilised to obtain many views and angles of the subject scene. This was utilised to increase data input sample size, to maximise the fidelity of the final reconstruction.

SfM and then PSR software was then utilised to produce a point cloud of a smoothed surface that can be used as input to the next stage, mimicking the kind of surface generally produced from high density Lidar data. Firstly, the collected video was split into individual frames via a simple Matlab script.

The resultant images can then be fed into the open source software Visual SfM (VSfM) [62, 63], to produce the sparse point cloud representation of the scene, which is highly representative but also with a high level of noise. Densification of this sparse cloud within the VSfM application, is achieved through use of the Multi-View Stereo (MVS) software plugin, PMVS2 [23]. This results in a visual representation that is even more representative of the scene but still with a high noise factor.

The final stage in the application of pre-existing software in the toolchain is the use of the open source Meshlab application. This is used to apply a Poisson-based surface tessellation to the result of densification, which itself produces a smooth surface, while mitigating much of the noise present after densification, and this surface can be extracted as a point cloud. This leads to an output that can be processed to extract fracture orientations.

1.4.2 Developing the Automatic System: GeoStructure.m

The resultant point cloud needed to be processed for anisotropic noise, that is an inevitable artefact of the SfM process. This was achieved through a Mahalanobis distance method, which accurately recognised structure within the point cloud, allowing a less arbitrary removal of noise, than with a standard Least Squares methodology.

The de-noised point cloud was then segmented using a volumetric voxel-fit method, originally developed for use with much higher data density Lidar point clouds [24]. This characterised the surface topology by dividing the entire point cloud space into volumetric pixels (voxels), and moving through the entire point cloud space, fitting planar surfaces. This yielded a dimensionless size factor, ζ , that described the relationship between the volume being evaluated and the overall volume of the point cloud space.

A k Nearest Neighbour Region Growing algorithm was used to detect large-scale regions of self-similar orientation, for classification and extraction. This algorithm utilised two internal variables relating to the angular difference between voxel planes, θ , and the co-planarity offset between voxel planes, ψ . Together with k , these provide four variables that control the accuracy of fracture orientation extraction, and this variable space search is the subjects of the results section.

Following the region growing stage, planar representations of region areas were extracted from the region data, using a localised co-ordinate system (u, v, n), instead of the standard World co-ordinate system (x, y, z). To find the minimum perimeter of the extracted plane, the skew symmetric rotation matrix $\hat{\omega}$ was used, in tandem with the φ Golden Section Search, to find the optimum minimum sized perimeter for the plane in question.

Finally, in order to bypass the discontinuity present in the direction cosine method of fracture orientation extraction, a method was developed that utilises the relative difference between the normal orientation of a ground-truth surface and a measured surface. This resulted in a smooth, continuous and uniformly comparable score, used to examine how accuracy is affected, as values for the variable quantities ζ , θ , ψ , k are changed.

1.5 Contributions of the Thesis

The contributions of this work are thus listed below:

- Developed a high density sample rate strategy for the collection SfM data, too maximise resultant point cloud fidelity
- Developed the Software Toolchain as a whole by:
 - Integrating SfM and PSR software to produce a surface representation of Lidar type
 - Integrating a Mahalanobis distance methodology for anisotropic noise processing
 - Applied a voxel-fit methodology previously used on high density Lidar point cloud data
 - Detected large scale regions in point cloud topology using a k NN Region Growing methodology

- Created an accurate bounded planar representation of regions using a localised coordinate reference methodology and a φ Golden Section Search to find the optimum perimeter for each
- Tested accuracy of reconstruction using a “minimum of normals” method, as an alternative to the standard direction cosines method for obtaining fracture orientations

1.6 Thesis Overview

That concludes the brief overview of the motivations for this thesis. The layout of the remainder of this document are as follows:

Chapter 2 will provide a review of the literature in various areas relating to the subject. This will cover the various aspects of SfM, Lidar technologies, computer vision, image processing and the use of cameras and camera platforms, as they relate to Geology.

Chapter 3 will provide detailed background on the key components of the proposed Software Toolchain, as covered in the review section. In particular, under review will be SfM underpinnings, surface tessellation techniques for point cloud processing, and the trigonometric techniques used in geology to extract planes structure and make accurate measurements.

Chapter 4 will present the proposed algorithm, describe structure and the reasoning for design choices. Also listed will be the contributions of the work and explanations of those contributions.

Chapter 5 will present the results of the Toolchain testing, which found optimum values for each of the four variable quantities controlling toolchain output.

Chapter 6 will conclude on the Software Toolchain, again, indicate the contributions and outline future work to be undertaken in this area.

Chapter 2

Literature Review

2.1 Geological Photogrammetry (Structure from 2D images)

Photogrammetry is the science of extraction of the geometric properties of objects, determined from imagery, and was one of the first remote sensing technologies ever developed. In practice, with it is approximately as old as modern photography, it's very first examples dating from the mid-nineteenth century.

A simple example of this is as follows: assuming the scale of the image, s , is known, the distance between two points that are co-planar parallel to the photographic image plane, can be calculated. This is done multiplying the distance between the points by $\frac{1}{s}$.

These simple beginnings can be expanded upon through various avenues. As an example, stereo-photogrammetry allows the estimation of three dimensional coordinates of points on an object. Measurements of common points are made in two or more images, viewing from different positions. A ray (line of sight line) can be drawn from the camera position to the point in question in both images. The triangulation of these rays determines the 3D location of the point.

These methodologies find their way into modern geology largely in the characterisation of rock units and determination of rock discontinuities and their orientations. Key work in this area has been done by Post [46] and Kemeny [35], whose contributions can be found in much of the research that has, as part of its focus, the metric characterisation of geology, from remotely sensed sources.

2.1.1 Extracting Orientations: The Hough Transform

Post et al. [46] illustrates an automated method of extracting 3D orientation measurements from images using the Hough transform, combined with the Fisher distribution as a method extracting estimated joint orientations from photogrammatic data. The authors dispensed with image pre-processing; noting that the majority of such pre-processing produces a binary image, they focussed on developing a special parameterisation of the Hough Transform, the $\rho - \theta$ Hough Transform (RTHT):

$$\rho = x \cos \theta + y \sin \theta \quad (2.1)$$

where ρ and θ represent the respective x and y axis of the normal vector, passing through the image origin [46].

This allowed them to find and extract linear fracture trace features from sub-plots of the images, which gave accurate fracture trace dimensions, assuming that information on the orientation of the camera and distance from the feature are known. Previous to the development of this method, any detected fracture would span the entire length of the image when returned to normal space from Hough space.

2.1.2 Extracting Orientations: The Fisher Constant and Distribution

Dip Angle and Dip Direction are derived using the Fisher constant, K , as a measure of scatter about the mean orientation:

$$f(\theta) = \frac{K e^{K \cos(\theta)}}{2\pi(e^K - e^{-K})} \quad (2.2)$$

The Fisher Distribution Probability Density function, gives the probability per unit angular area of finding a direction centred at an angle from the true mean. This is a symmetric distribution and has a maximum at the true mean.

In a follow up paper [35], the authors describe a methodology for estimating 3D joint and orientation measurements from images, by delineating fracture traces based on some assumptions:

- Fractures occur in sets,
- Each set can be described by a mean orientation,
- Each set can be described by a measure of the scatter about the mean.

The authors continue to use the Fisher distribution to determine the trace angle distribution for each joint. The authors moved to a genetic algorithm to solve for the orientation measurements, rather than using a Monte Carlo simulation, as in the previous work.

Simulation data indicated a high level of accuracy from the method when applied, and the authors noted some interesting benefits even over some more advanced strategies. They cited improvements in safety and also appreciated the automated nature of the process, suggesting that human bias inherent in individuals making independent measurements by hand, could be more readily mitigated. The authors noted that 3D fracture surfaces are difficult to render on smooth rock faces and that analysing fracture surfaces using stereo imaging, for instance, involves many more measurements that can result in error.

2.2 Structure from Lidar and Light Ranging Technologies

Extraction of orientation measurements from images leads to inherent issues with accuracy and fidelity. These can be addressed through the use of the laser scanning and Lidar technologies. Both of these employ the use of laser light to range an object from distance. This leads to extremely high fidelity datasets, incorporating orientation and potentially other data, such as spectroscopic.

2.2.1 Demonstrating the Total Station

Advances in sensing technology led to the application of new technologies in the area of structural geology. Within a similar time to when Kemeny and Post produced their work on photogrammetry, Feng et al. [19] demonstrated the use of the Non-Reflector Total Station (TS) in the acquisition of orientation measurements, in a geological setting. A TS is a device that can capture the spatial positions of target points on a surface, and operates in a way comparable to Lidar.

The authors successfully demonstrate data acquisition and calculation of Strike, Dip Angle and Dip Direction fracture orientations, through the determination of the ideal (best-fit) plane of the surface and use of the *direction cosine* method; using the vector normal of that plane. This basic method for deriving orientations is referenced in various work in the literature [24, 59].

The authors list the following as benefits over the standard compass-inclinometer method:

1. The surveyor can take orientation measurements without physically touching the rock face,
2. The problem of apparent orientation measurements can be reduced because the deepest inclination of the fracture plane can be calculated by the co-ordinates of a number of points, not decided by visual examination,
3. Since orientation measurements taken by the TS method are stored in digital form in a computer, any further analysis of the features of fracture sets can be easily conducted.

2.2.2 Reconstructing on a Larger Scale

During this time, research was progressing in the reconstruction of large scale scenes from point data. Hu et al. [30] successfully demonstrated reconstruction of a complex townscape scene from aerial Lidar data. It focussed on the use of geometric primitive shape models to accurately reconstruct buildings and complex structures, during the point cloud segmentation process. This work is of importance as the accurate extraction of structure from large scale and/or noisy point datasets is a problem that needs to be solved, to progress in this area.

Only a couple of years after the work of Kemeny and Post in the area of geological photogrammetry, Belian et al. [2] applied terrestrial laser scanning to outcrop stratigraphic mapping. High fidelity and high accuracy spatial and light intensity points were used, acquired via Lidar sensing to conduct surface characterisation and construct 3D geological models. They use outcrop data to explain their devised workflow and discuss the building of their 3D Digital Outcrop Models, focussing particularly on reproducibility and quantification.

They come to some interesting conclusions about the potential of a system such as this in geological work:

1. Lidar can be used to gather essential three-dimensional point information and make geological interpretations directly on them,
2. Oblique outcrop geometries can be projected onto a specific plane efficiently enough to get true strike or true dip perspectives, preserving scale and geometry,
3. Field trips can now begin at the office to prepare participants for what they are going to see, highlight key observations to note, and preview potential safety hazards,

4. Field geologists can use these data to more accurately estimate budgets for field expeditions,
5. Areas traditionally inaccessible can be brought back into the office, classroom, or visualization laboratory [2].

2.2.3 Automating Orientation Extraction

Investigating the potential for automation of geological analysis, Gigli et al. [24] developed an integrated piece of software to semi-automatically analyse and derive orientation measurements from the Lidar point clouds of geological formations. The software tool called **DiAna** (Discontinuity Analysis), was designed to perform both 2D and 3D structural analysis of rock discontinuities from high resolution laser scanning data.

The segmentation of structure in the 3D point cloud was achieved by dividing the entire point space into a grid equally sized volumes (*voxels*), and moving a *search cube* through each of these spaces to evaluate the points inside. Evaluation consisted of applying a least squares calculation to the space; using this to generate a best fit plane for that volume. A measure of standard deviation was used to validate the plane against the rest of the surface, and accurately detect structure at various scales. Orientations were derived using a direction cosine methodology very similar in type to Feng et al. [19].

This demonstrated direct fracture orientation measurement from point data, comparing for accuracy well with traditional methods of evaluation in this area. Six of the ten descriptive properties of discontinuities (orientation, spacing, persistence, roughness, number of sets and block size) can be semi-automatically extracted using this method, with the other four (aperture, seepage, wall strength and filling) necessarily requiring physical contact with the surface.

2.3 Structure from Motion

Structure from Motion (SfM) as an image processing technique, extracts structure from sequences of images that, by their relation to each other, suggest localised motion.

As a subset of Multi-View Geometry (MVG), finding the structure from motion is a similar problem to finding structure from stereo vision, both cases centering on deriving an independent object in 3D space from a planar representation.

2.3.1 Scale Invariant Features

Although there were certainly methodologies to accomplish this before [27, 10], significant progress in this area was made by Lowe [40, 6, 39], with his work on object recognition from scale invariant features.

The author created the Scale Invariant Feature Transform (SIFT) methodology through the development of *key descriptors*, that use a Difference-of-Gaussians (DoG) pyramid maxima to detect significant features. The descriptor is rotated to the dominant gradient direction in order to make it invariant to scale. The four key parts of the SIFT algorithm are detailed as follows:

1. **Scale-space extrema detection:** A search over all scales and image locations. It uses a Difference of Gaussian function to identify points of interest invariant to scale and orientation.
2. **Keypoint localization:** At each location, a model is used to determine location and scale. Keypoints are selected based on their stability.
3. **Orientation assignment:** Orientation(s) are assigned to each keypoint location based on local image gradient directions. Operations for each feature are only performed on data that has been transformed relative to the assigned orientation, scale, and location. This is the step that provides invariance to transformation for the features.
4. **Keypoint descriptor:** Image gradients are measured in the region around each keypoint, which are then transformed into a representation that allow for change in local shape and local lighting.

Broadly, the entire SIFT algorithm works by recognising points in *scale space*, creating image keys that allow for geometrical deformation at multiple scales, and, that are invariant to image scaling, translation, rotation, partially invariant to lighting and affine or 3D projection. Lowe also demonstrates that the algorithm can detect unique objects that are partially occluded, and, groups of similar objects that are occluded and/or translated relative to each other. This laid the foundation for much of the future work in the area and this, and/or similar algorithms form the core of any SfM system.



Figure 2.1: An example of a sparse point cloud, with the individual view positions and orientations visible (the snake-like structure).

2.3.2 Automated Bundle Adjustment

This was followed up with further work [6], attempting to fully automate the image feature matching process. Focussing on combining the Scale Invariant Feature Transform (SIFT) and Random Sample Consensus (RANSAC) [20, 39], with parameterised camera matrices and a sparse bundle adjustment algorithm, this contribution demonstrated object reconstruction, from unordered image sets with no user intervention at all.

The algorithm took the following form, assuming that the object or scenes in question are rigid and that the images are obtained using perspective camera:

- Associate a camera matrix to each image, this is parameterised by rotation, translation and focal length.
- Find matches between all images using invariant local features
- Use RANSAC to filter for those matches consistent with the fundamental matrix
- Define objects of subsets of matching images

- Apply a sparse bundle adjustment algorithm to solve for the structure and motion of each defined object, producing a sparse point cloud

The remarkable aspect of this work is the ability of the complete algorithm to fully distinguish between images of two or more objects, or scenes in the same input dataset. This combined with the other developments came together to further progress the core form of the SfM algorithm, adding both a level of robustness and high degree of automation, making the technology more practically usable.

2.3.3 SfM Pipeline Synthesis

With the progress made in previous work [40, 39, 7, 6], all the key components practical development and application of end-to-end SfM pipeline/toolchain systems was now available. A wide variety of applications have been since utilised, several of which were in the area of architectural or landscape reconstruction, for various applications. One of the most widely known was Snavely et al. [52], who concentrated on the reconstruction of architectural scenes. The authors developed a system incorporating model-based front-end. The front-end has an interface that automatically computes and displays the viewpoint of each photograph, as well as a bundle adjusted sparse point cloud of the scene.

This contribution also allows the rendering of each image position in the scene, along with exploration of the space generated by the SfM algorithm, allowing easy, accurate reconstruction of full locations. The author noted several areas of deficiency, many of which would be further developed or solved by future work:

- A more intelligent image registration process would lead to processing time performance increases by allowing a minimum number of views confirming overall scene structure, with remaining views optimising scene detail
- Inability of the camera model to be able to mitigate lens distortion
- The need for Ground Control Points (GCPs) to produce a parametrised scene reconstruction

2.3.4 SfM: Continuing Refinement

As SfM centric MVG was being applied by bringing together all the necessary component parts, the details of how to obtain the best results from such a system were being explored. As an example, Goesele et al. [25] re-explored some of the earliest window-based reconstruction methodologies, constructing only areas with a high degree of certainty, rather than employing an approach that enforces completeness throughout the reconstruction. Despite the inevitable likelihood of holes occurring in the shape model, the author found a reconstruction completeness and accuracy on par with the most commonly used methods of the time.

Other work focussed on using video for structure from motion and putting more serious consideration into the algorithm work flow. Frahm et al. [22] used SfM to reconstruct architectural scenes from video and the authors demonstrate several algorithms to improve the workflow and/or output of several parts of the process, including GPU algorithms for image feature matching and the viewpoint invariant patch (VIP) algorithm, which extended the wide baseline features that formed part of the SIFT algorithm [40].

The work expands upon the standard process with extensions to sparse bundle adjustment, dense model generation, [image] feature matching and camera pose estimation, which they approach with a real-time algorithm, robust to outliers and noise, that proved highly effective under time constraints. The authors conclusively state that a system such as this can be set up and used to take both relative and absolute measurements, given a few variables. Despite the various successes of their work, they encountered a significant barrier in terms of processing time performance, when attempting to process a scene of more than a certain size, suggesting that the solution lay with improvements to algorithmic and/or computing architecture.

2.3.5 Densification

Progression in the area of SfM has now produced the *sparse point cloud*. This is a representation, but objects reconstructed to this point lack solidity and detail. This allowed for several solutions, focussing on creating accurate representations through various techniques, such as surface meshing and tessellation.

Furukawa et al. [23] produced one of the most novel and accurate solutions to the densification problem. This work proposed a new method of densifying sparse point clouds by matching features across multiple images, and using these to form structures called *patches*.

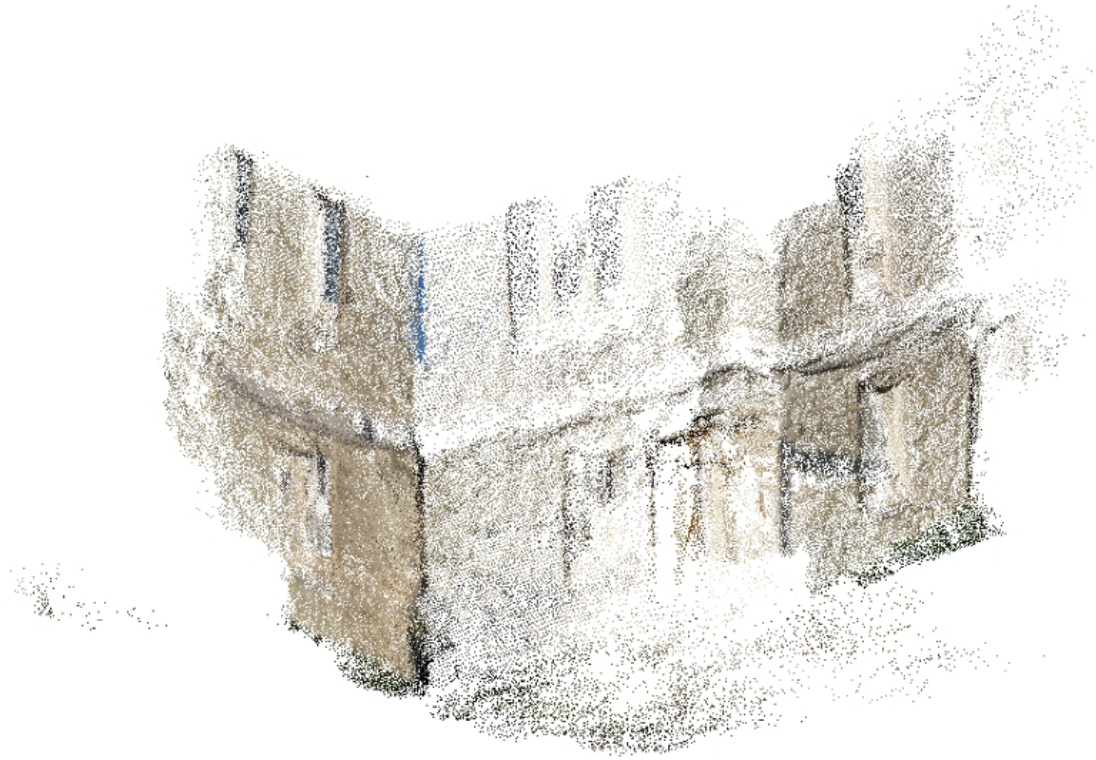


Figure 2.2: An example of the result obtainable through the densification of a sparse point cloud using the Furukawa patch-based method.

A patch is defined as a local tangent plane approximating a surface, with its geometry fully defined by its centre, unit normal vector (oriented directly toward the camera centre) and a reference image, in which the patch is visible. Similar patches are then matched and expanded using keypoints, filtered using various visibility constraints, and subjected to an expansion procedure to obtain a denser set of patches, before using visibility constraints to filter away false matches. The resultant structure can be meshed into a solid representation using a tessellation technique; the authors elected to use the Poisson Surface Reconstruction (PSR) tessellation, which has many interesting features, such as the characteristic reconstruction of very smooth surfaces, and, the ability to comfortably handle very noisy and “imperfect” data. It is regularly used in the literature [23, 33, 34, 8, 53, 18, 58] to derive solutions in various problem spaces.

The final algorithm showed a consistently advanced level of accuracy, compared to alternative techniques along with the following characteristics:

- It can handle a variety of datasets and allows outliers or obstacles in the images,
- It requires no assumptions on the topology of an object or a scene,
- No need for any type of initialization, such as a visual hull model or a bounding box, but

can use such information when available,

- Consistently high performance on objects with fine surface details, deep concavities, thin structures, outdoor scenes with restricted views, and scenes where moving obstacles surround a structure of interest.

The results of this work were significant and the freely available algorithm forms the final part of many available SfM toolchains, usually as a plug-in or software extension.

2.3.6 SfM: Utilising an End-to-End Framework

The generalised framework exists to utilise SfM for research in the area of computer vision (CV) and passive reconstruction. Falkingham [18] discusses the application of freely available and open-source software, in the production of high quality point clouds for various applications, focussing specifically on the reconstruction of palaeontological samples of various scales. Even while various new uses were being demonstrated, research [11, 10] was still demonstrating that improvements and new capabilities were possible in various aspects of the technology.

Continuing development of the components, and usable solutions for those not expert in the development of computing algorithms or software frameworks, lead to the development of several commercial and non-commercial software [18, 59, 3, 56, 41, 42], although many lacked either the universal plug-and-play usability, were expensive (if commercial) or consisted of many different component parts that lacked reliability.

A complete semi-automated geological evaluation system using SfM from photogrammetry and a drone for image acquisition is demonstrated in recent work [59], with the authors developing a semi-automatic method for measuring joint orientations from geological photogrammetric data processed using structure from motion techniques, into a point cloud. RANSAC was used to develop orientation measurements of surfaces from the point cloud and a standard geometric method to derive the Dip Angle and Dip Directions from those measurements. These were then compared with ground-truth data from human measurement.

For data acquisition, the authors acquired approximately 140 photographs using a drone at an altitude of 3040 m in Tasmania. A five minute low altitude flight resulted in high resolution photographs (approx. 1 pixel/cm), covering a 100 m by 80 m. The resulting point cloud, was created using SfM techniques, following proven methodologies [28, 57]. The sparse point-

cloud was created using a Bundler software package [52]. Feature points were identified via SIFT [40, 39, 6], and matched using nearest neighbour kd-trees. Poor matches were filtered via RANSAC, and final 3D calculations were done through bundle block adjustment, producing a sparse point cloud. This was used for input to Patch View Multi-Stereo (PMVS2) software [23], and a dense point cloud was created.

Finally, the point cloud coordinate system was transformed to the real world coordinate system and a digital elevation model (DEM) generated. The individual photos were geo-referenced and stitched together to generate a single mosaic [57]. This produced a relative accuracy of approximately 1025 cm.

The authors tackle the issue of rock feature detection through the use of image edge detection methods, primarily phase symmetry/congruency and canny methods. The the images are put through segment linking (anti-segmentation) process in three stages:

1. Morphological dilation followed by thinning to fill in small gaps between close features. The endpoints of feature segments were identified and dilation was performed using a structuring element only at the endpoints to connect nearby features together.
2. Line Segment Categorisation. Next line segments that lie near each other are connected based on a probability calculation of being the same structure. A modified Hough transform [46, 35], that is a widely used method for extracting dominant lines in edge images is used to find the probability.
3. Line Segment Linkage. Segment linkage proceeds by an automated algorithm. The algorithm identifies the longest feature segment and the segments sharing similar orientation and these are linked together if their alignments meet certain criteria.

The authors also produced a semi-automatic option, guided by user inputs. This allowed human interpretation by displaying the segment detection result and the features being selected with the mouse.

Only after this do the authors move onto deriving the 3D structure by calculating the z coordinate of the individual features, using bilinear interpolation. This requires orthorectified images or camera location, and calibration parameters. The authors then use RANSAC, the plane equation and inverse cosine geometrical identity to derive the final measurements.

There are drawbacks with Vasuki's methodology that this work seeks to address: The method required user input to identify rock discontinuities, by manually clicking on the original im-

ages, using this as the input to transform to a 3D representation. It is asserted in this monograph, that this step can be dispensed with, by relying on the accuracy of SfM methods. Increasing the number of unique views, therefore angles, can improve the fidelity of the final reconstruction (with some intermediary steps), well enough to extract measurements of scientific quality from it. This work also seeks to move to a *fully automated* [SfM-based] system, completely removing the need for user intervention at any stage of the process.

2.3.7 SfM: Developing Plug-n-Play Solutions

The work of Wu produced one of the most utilised and well-known examples of Plug-n-Play SfM software, being open source, free and simple to use [62, 63]. Visual SfM (VSfM) was developed in the course of examining the efficiency of the sparse bundle adjustment and in particular, the role of hardware optimisation in the process. The authors developed a new implementation of Newton-type bundle adjustment that makes use of parallel processing in the solving of large scale reconstruction problems, exploring parallelisation of the problem using multi-core CPU and GPU hardware architectures in the process.

The author showed dramatic improvements of upto ten times for CPU's and thirty times for GPU's, while maintaining comparable convergence behaviour. Follow up research [63] continued to improve on the SfM algorithm, this time concerning efficiency of the Feature Matching part of the algorithm. The author demonstrates improvement to the Feature Matching aspect of the SfM algorithm as a way of mitigating the geometric time factor that occurs with Feature Matching. Wu demonstrates that the processing problem for this stage is of $O(n^2)$ time complexity, but only needs $O(n)$ time on its major steps while still maintaining the reconstruction accuracy [63]. The authors demonstrate a practical run time of $O(n)$ for large reconstructions, as long as the number of independent views is less than 15,000 cameras.

The result of this work was the release of Wu's VSfM software application, integrating the various steps in the SfM algorithmic chain (SIFT, Feature Matching, Bundle Adjustment), and developments in the SfM CV area (BA from unordered image sets, software/CPU/GPU parallelisation) into one easy-to-use program that enables the generation of a high fidelity sparse model from an image set, in the majority of circumstances.

It is around this time that SfM applications started being favoured in geological work as a supplement to, or replacement for, traditional photogrammetric methods in general aspects of surveying. Several aspects show promise in comparison to other methods and these points have

been commented on by several researchers.

2.3.8 Integration of SfM and Geology

Westoby [61] demonstrated the effectiveness of use of SfM derived Digital Elevation Models (DEMs) in geoscience, using a terrestrial laser scan for comparison. Across a variety of topographic types, this work showed that SfM DEMs are quite capable of decimeter scale accuracy, a sentiment echoed by other researchers [27, 28]. This was shown to be even in the case of complex topographies, as long as the image set consisted of images from a wide array of positions, with a high level of overlap between them.

The application of this method for more intensive geological study in the same year by James et al. [31], who applied this to the analysis of surface processes, such as mass erosion and movement. The author looks at sample data from across a wide variety of scales, using several metrics to evaluate the effectiveness of the SfM techniques against traditional methods. The authors note a general precision ratio of centimetres over 10's of meters and successfully use these methods to monitor erosion rates along a stretch of coastal cliff, generating results that, when compared with a laser scanner survey of the same site, produced comparable data and reduced data collection time by similar to 80%, while being far easier to use than traditional photogrammetric or laser scanning methods.

2.3.9 Data Acquisition Strategies: Exploring Camera Paths

More recent examinations of the technology have focussed on improving accuracy of the technique using a variety of conditions. DallAsta [15] used a Monte Carlo based evaluation protocol, to test the differences between view configurations; a straight line through a scene, or a circular path with all views converging on the same area. They also examined several other areas including the use of ground control points, autofocus and camera calibration on performance.

However in a follow up to previous work, James et al. [32] come to the firm conclusion, that relying on automatic camera calibration can lead to systematic errors in final results, but also develop several other methodologies to mitigate these effects. Of particular interest is the effects of camera path, and the addition of control points so that radial distortion can be adjusted for. If a nadir image set is not required, then flying overlapping flight lines in opposing

directions with an off-nadir camera, is an effective method of mitigating distorting camera perspective effects common in uncalibrated SfM datasets. The addition of oblique images is a simple way of rectifying the problems with radial distortion in images leading to deformed and skewed point clouds [15, 3, 31, 59].

2.4 Point Cloud Processing

Once acquisition of point cloud data has occurred, post-processing extracts and indicates structure. It is possible to use numerous image and data processing techniques that will all result in the *segmentation* of a point structure, into some kind of solid surface shape.

The work of Gomez [26] utilised a method of evaluating surface roughness in high density point clouds by segmenting the total volume into voxels that become progressively smaller as more detail is evaluated using a scalar complexity metric.

2.4.1 Approaches to Segmentation

Work in this area by Cohen-Steiner [12] considers the problem of efficiently capturing the geometry of a surface and presents a contribution, that results in a framework for generation of surfaces using approximations of surface shapes, that is novel and flexible. The authors indicate that their approach is entirely discrete and error driven, not requiring parametrisation, or local estimations of differential quantities. Along with a new metric based on normal deviation, they show excellent results at capturing anisotropy in such structures.

Random Sample Consensus

RANSAC [20] is regularly applied to various solutions in this area because it has the specific characteristic of interpreting and smoothing data, containing a significant number of gross errors (such as a noisy point cloud). This leads to many implementations of RANSAC, when considering structure extraction of this kind [20, 63, 22, 17, 52, 3, 49, 9, 6, 24].

Schnabel [50] demonstrates this, with a method recognising several shape primitives within noisy point cloud data. The RANSAC Sampling method deconstructs the point cloud into a structure of shapes, along with any remaining unclassified points. The algorithm is robust

even in the presence of high density outliers and a high degree of noise. The method scales with point cloud size, the shape number and size within the data. Even point sets with several million samples are robustly decomposed within less than a minute.

Tessellation and Meshing Techniques

Delaunay tessellation techniques [13] also proved useful in this area with hybrid tessellation minimally covering the proximal space around each point allowing for particularly efficient meshing on a scattered point set. Along with the Delaunay method, the PSR method [34] is a popular method for surface reconstruction from point clouds. PSR works by applying convex hulls to point sets.

The authors illustrate a new method for applying surface reconstruction to the point sets, by extending the technique to explicitly incorporate the points as interpolation constraints. The extension also allows geometry processing techniques, the screening term is defined over a sparse set of points, demonstrating that high density clouds are not required for this technique and that sparse constraints can readily be integrated efficiently.

The work of Pauly et al. [45] devised a new method for smoothing and meshing surfaces on point cloud structures. Specifically focusing on the processing of simplified clouds, several methodologies are developed to down-sample a point cloud, simplifying its structure without losing detail or requiring an intermediate tessellation stage.

Techniques Incorporating Region Growing

Rabbani's investigation [48] into this problem demonstrated a comprehensive method of structure extraction, bringing together several techniques, to provide a framework for a fully automatic solution for this process. The authors devised a full algorithm for automatic segmentation of point clouds using a k Nearest Neighbour Region Growing algorithm, to find surface planes. Accurate surface characterisation is achieved using surface normal calculation, a smoothness constraint, and a residual measurement, that denotes relative curvature. As a segmentation algorithm, the whole process is effective and forms part of the basis for the implementation of the Toolchain, in this monograph.

Vosselman et al. [60] developed a method of shape extraction from point cloud using the Hough Transform. Used extensively in various type of image processing and already referenced in

several works examined in this review, authors utilise the 3D variant for the extraction of planes from the point data, trialling a method that used plane normals and method that does not. For scenes of very high complexity (≥ 20 million points), surface [region] growing techniques were used for point cloud segmentation. Region Growing (RG) extended the categorisation of the surface point to adjacent point, using three criteria:

1. Proximity, where only points that are near the surface point is added to the surface,
2. Locally planar, the plane equation is used to extract the correct orientation for that point,
3. Smooth normal vector field, to enforce a smooth surface, a candidate point is only accepted if the angle between its surface normal and the normal of the nearest point is within a threshold.

The authors used the Hough Transform to extract man-made shapes from an industrial scene and, extended the method by utilising a combination of the transform and region growing to reconstruct a large cityscape.

2.4.2 Filtering Noise and Erroneous Points

When dealing with point clouds, especially software generated point clouds (SGPC), noise can be a major concern regarding the accurate extraction of structure. Several of the following works in the literature illustrate methods of mitigating this. Both non-parametric and parametric methods exist. Xi [64] used a non-parametric method incorporating an anisotropic kernel density estimation for outlier removal, along with a hill climbing line search for accurate approximation of the real surface boundary.

Brophy [5] successfully asserts that a Mahalanobis distance implementation can remove outlier and noise points from a preprocessed point cloud, revealing the solid surface structure, with high fidelity. His technique applies a Mahalanobis distance, within an iterative Parzen Window search across the point cloud. This achieves good anisotropic structure recognition, although is only applied to clouds treated with a Gaussian noise distribution.



Figure 2.3: A commercially available drone, fitted with a fixed orientation 14 Mpixel fish-eye camera.

2.5 Drone Platforms and Aerial Photogrammetry

The use of Drones/Unmanned Aerial Vehicles's (UAVs) as a survey tool has been a prevalent idea for several years now. They provide an effective method of data acquisition from normally inaccessible areas, thus providing data from a different perspective. The decrease in cost, size, and, the increase in capability and fidelity of camera devices, combines with the flexibility of the drone to create an excellent platform for data and image acquisition.

As with many technologies, the first places they find use is in situations of extreme danger, where a different perspective is needed, and/or where human intervention is particularly difficult or troublesome. This technology has been shown to be useful in many areas, such as disaster management [1], civil planning [36], and of course, military applications. Geological Science has found particular use for the technology, especially concerning the area of photogrammetry, which, as mentioned earlier in this Section, in the field of geology is particularly concerned with the deriving of 3D structure from imagery [46, 35].

2.5.1 Applications in the Field

The use of drone technology for geological structural characterisation and survey has become widespread. A good summary of the development of this trend can be found in Bemis' 2014

research [3]. This paper considers the techniques that can be used for high fidelity surveys of trenches, rock exposures and other features, to evaluate potential for paleo-seismology and structural geology.

Strategies for Data Acquisition: Flight Plan

The authors devise and explain the general approach they employ, acquiring data and processing, describing strategies for both ground-based DSLR cameras, and imagery obtained from drone cameras. They also list various tools available for use, with some description as to the general working of the different software.

The authors highlight various factors contributing to survey success in their conclusions, with particular interest being obtaining the best resolution and accuracy for any survey. Indicating work by James and Robson [31, 32] as a good guide in the planning of an appropriate survey structure for the best measurement precision, stating that millimetre scale precision requires photographs to be collected within several meters of the target, in addition to requirements for well-distributed camera positions and Ground Control Points (GCP).

This is of particular relevance, indicating necessary considerations in implementing the best possible collection strategy of initial data, and continues to reinforce the idea posited by other researchers, that this will have a definite effect on the quality of measurements obtainable from the software point cloud at the end of the process [3, 59, 56].

Strategies for Data Acquisition: Number of Views

As mentioned previously in this section, the inclusion of as many angles of the object/scene as possible is highly important for point cloud reconstruction with fidelity. The introduction of systematic errors into the reconstruction from the image acquisition process is something that James et al. [32] examines. This work investigates the mitigation of the systematic errors that are introduced through the acquisition process. Specifically, the authors reference the parabolic-type distortion effects that can occur in the SfM process, which finds expression as “doming” of a surface, and conclude that this results from a combination of near-parallel imaging directions and poor correction of radial lens distortion.

The addition of oblique images [59] to any dataset is an effective way of tackling issues regarding radial distortion in images, which would lead to deformed and skewed point clouds. This

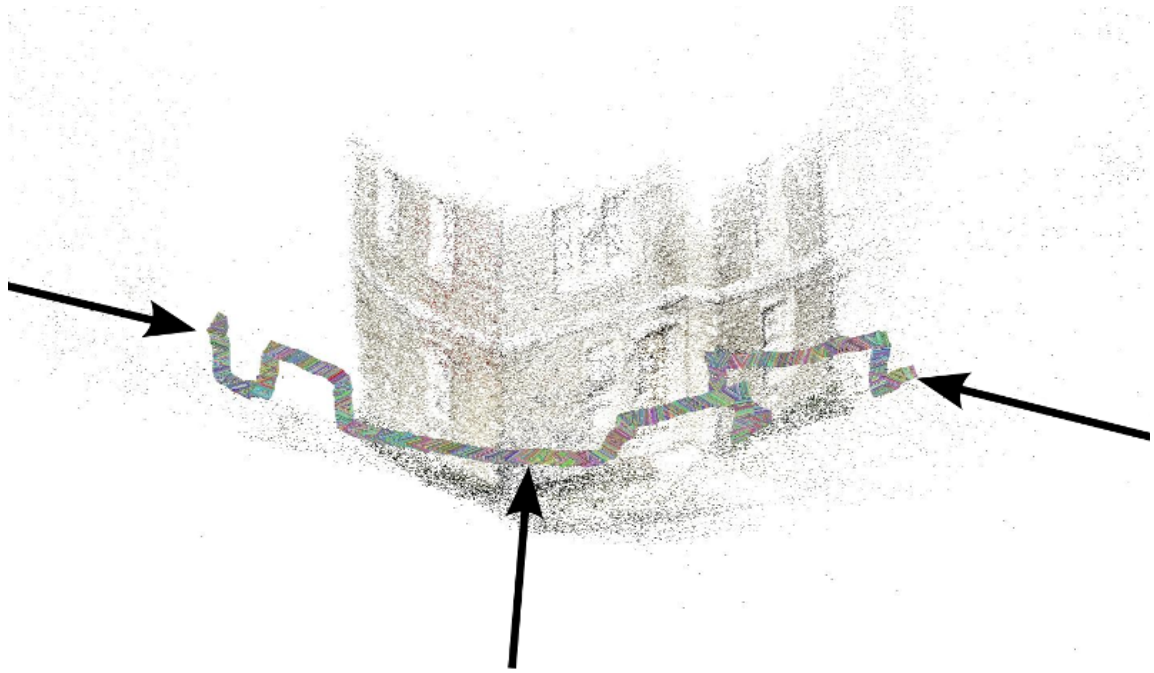


Figure 2.4: The bundle adjustment process reconstructs not only the subject but also the positions of the views used in the reconstruction. The flight path of the camera used is visible and highlighted in this example. The high density of view samples is also apparent, being as it appears as one continuous line.

focus' the discussion back onto the concept of flight plan.

It is interesting to consider whether in this case more unique views of the object or scene should enable a higher fidelity point cloud structure. Obviously, the way to achieve this in an SfM context is to have a higher frequency of image collection. The simplest way to achieve this is to use video, rather than images, then a very high frequency 30 frames per second (fps) data acquisition can be achieved, with the flight plan only being the consideration, rather than flight plan and image acquisition frequency, which must be also actively implemented for a set up where a non-video camera is utilised.

McLeod [44] takes this approach, with work on measuring fracture orientation in an open mine. The author uses video from a drone to measure fracture orientations from a sparse point cloud. The author uses a combination of various commercial and open source software to constitute her pipeline.

2.5.2 Integrated Methods in Geological Analysis

Very recent work by Chen [9] shows that drones can be utilised effectively as part of an integrated approach in structural geological analysis. This work combined terrestrial Lidar scans with drone based Digital Elevation Models (DEM) and satellite data. From drone images a DEM was derived and used to find sediment orientations. This constrained interpretations of seismic data. Terrestrial Lidar was then used to generate high-resolution digital outcrops and to evaluate the accuracy of the drone-based DEM.

Again, RANSAC was used in post acquisition processing, to derive the orientation planes of the stratigraphic bedding, through this demonstrating that drone based data can be used to evaluate geologic structure, and, confirming that more comprehensive analysis of the structural geology is possible using this methodology.

2.5.3 The Georeferencing Problem

Much of the work regarding data obtained from drones has concerned geo-referencing and obtaining an accurate location for the data in particular. Harwin et al. [28] considers methods of assessing the accuracy of of MVG point clouds from drones. Here, the authors used photogrammetric monitoring for a coastal region in Tasmania, and compared this to a similar protocol for TS. They were able to utilise full resolution imagery in their pipeline, comparing point clouds to TS survey data to assess accuracy. The authors determined georeferenced point clouds accurate to 2540 mm, obtained from imagery acquired at 50 m, again, confirming the high level of precision possible from these techniques.

Further work specialising in this area, produces some interesting results to consider. Turner produced two interesting investigations [57, 56], both of which, illustrated or developed useful techniques in the processing of SfM point clouds in the geological context. Work in 2012 developing an automated technique for generating georectified mosaics based on SfM point clouds from ultra-high resolution drone imagery.

Imagery of moss beds in Antarctica was used to validate the spatial accuracy of our technique which can directly geo-reference the imagery or use GCPs if they are available. Two datasets (one containing 200 images, the other 69 images) were processed with both techniques producing four mosaics. The directly geo-referenced mosaics had a spatial accuracy of 65120 cm with whilst the GCP technique achieved a spatial accuracy of 1015 cm.

As a progression from the previous work in this area, the authors developed a geo-referencing system [56], that requires no pre-defined ground control points. The authors compared three sparse reconstruction methods, comparing the geo-referencing steps against each other, in categories such as ease-of-use, processing speed and accuracy of spatial result.

Chapter 3

Theoretical Background

This section will focus on specific work, from which elements can be implemented, providing the reader with a detailed theoretical background knowledge of the areas utilised in development. Each section will borrow heavily from particular articles of research and these will be highlighted as necessary.

3.1 Multi-View Geometry: Theoretical Underpinnings

Crucial to the reconstruction of spatial (3D) scenes from a planar (2D/image) representation is the integration of multiple views and angles. This simulates motion, which then approximates the stereo vision depth perception effect common in biology. Unlike Lidar, which builds a representation of the scene or object through the recording of the physical positions of the contact points of a laser beam, an MVS/MVG/SfM algorithm must build its representation from a model developed, by image processing of acquired samples (images).

3.2 SfM: SIFT

After data acquisition, the first step in this processing chain is *Image Feature Matching* [40, 39, 62, 63, 6] and this is a key component of the SfM algorithm. This requires the identification and characterisation of as many of the image “features” as possible; a process that is then repeated for all images in the set. The aggregate effect is that the unique features shared between images

can lead to region stability between views, and therefore, recognition.

The two main algorithms used for this purpose are Scale Invariant Feature Transform (SIFT) and Scaled Up Robust Features (SURF). This work will focus on the SIFT algorithm, which was the first of the two to be developed, and a step change in effectiveness for object recognition algorithms in computer vision. The following sections heavily reference the various works of Lowe and others [40, 39, 62, 63, 6], in this area.

3.2.1 The Features of SIFT

Lowe’s [39] work develops a methodology where the geometric cost of processing images is minimised by taking a cascading filtering approach, in which expensive operations are applied only at locations that pass an initial test. Lowe separates the task into four distinct stages:

1. Scale Space Extrema Detection,
2. Keypoint Localisation,
3. Orientation Assignment,
4. Keypoint Descriptor.

What follows below is an abridged technical description of each of these areas, and these descriptions mostly borrow from the description of Lowe et al.

Scale Space Extrema Detection

Scale Space Extrema Detection (SSED) relates to the detection and characterisation of “key-points”. Keypoints are small regions inside detected features, and their detection requires identification of a region that can be repeatedly processed for different views and scales. This is done by utilising a continuous function called *scale space*, to search for features stable across all scales [40, 39].

It has been deduced through previous work [39], that only the Gaussian function can act as a scale space kernel. Thus, the scale space of an image is defined as a function, $L(x, y, \sigma)$ that is the result of a convolution of a variable scale Gaussian $G(x, y, \sigma)$ with an input image $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.1)$$

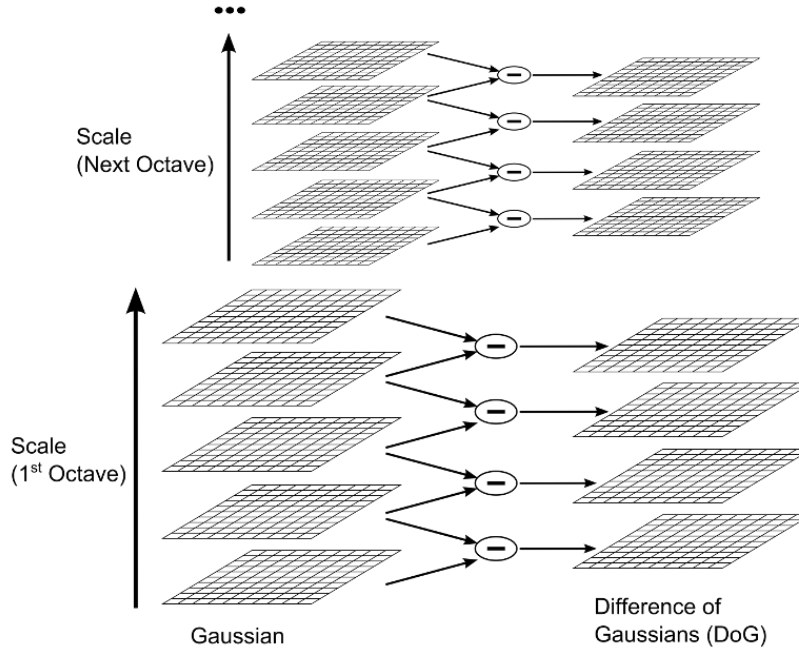


Figure 3.1: The Difference of Gaussians operation, as demonstrated by Lowe. A process of progressive subtraction, from an initial image $I(x, y)$, incrementally convolved with Gaussian functions, $G(x, y, k\sigma)$

where $*$ is the convolution of x and y and the Gaussian function expands to

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (3.2)$$

Lowe uses the *scale space extrema* present in the Difference of Gaussians (DoG) function to extract stable keypoints. The product of DoG operation: $D(x, y, \sigma)$, is resultant of the difference between two scales, separated by a constant multiplicative factor k :

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (3.3)$$

$$= L(x, y, k\sigma) - L(x, y, \sigma) \quad (3.4)$$

There are various advantages to the use of this methodology, however that discussion is beyond the scope of this monograph. It has been thoroughly examined in previous work [40, 39, 23, 62, 63].

SSE Practical Application

What follows here is a practical explanation of the SSE detection process as described. This particularly useful description is as described by Lowe [39].

The initial image $I(x, y)$, is incrementally convolved with Gaussian functions, $G(x, y, k\sigma)$. This produces images separated by a constant factor k in scale space, $D(x, y, \sigma)$. Each doubling of σ (referred to as an *octave*) in scale space, divides by an integer number, s , of intervals, such that $k = 2^{\frac{1}{s}}$. If $s + 3$ convolved images is produced for each octave, the extrema detection spans the entire octave. Adjacent image scales can then be subtracted to produce the DoG images. Once a complete octave has been processed, every second pixel in the Gaussian image, with a sigma twice the initial σ is sampled and the process is repeated. The final part of this stage is the detection of the SSE itself. The explicit aim here, is to extract the local maxima and minima of $D(x, y, \sigma)$. Each sample pixel in the current image is compared to its eight neighbours, plus the nine nearest pixels in the scales immediately above and below. Pixels are classified as extrema (keypoints) and selected only if their height is larger or smaller than all neighbours.

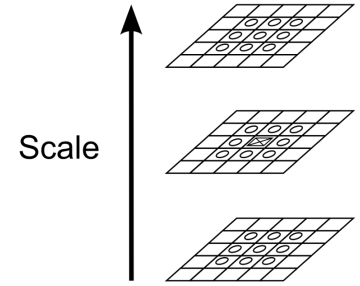


Figure 3.2: The scale space extrema detection operation. Pixels are compared to see if they are the local maxima or minima of $D(x, y, \sigma)$

Keypoint Localisation

Once a keypoint has been selected, it must be fitted to local data for location, scale and principle curvature comparison [39]. The end result allows the rejection of points based on low contrast (susceptible to noise) or poor localisation along an edge.

This is done using a 3D quadratic function, which is applied to the local sample pixels to determine the maximum [39, 6]. The approach uses the Taylor Expansion of the scale space function $D(x, y, \sigma)$, modified to have the origin at the coordinates of the sample pixel:

$$D(\mathbf{x}) = D + \frac{\partial D^\top}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} \quad (3.5)$$

where $\mathbf{x} = (x, y, \sigma)^\top$ and is the offset from this point. The location of the extremum $\hat{\mathbf{x}}$ is found by zeroing the derivative of this function, with respect to \mathbf{x} [39]:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}} \quad (3.6)$$

The Hessian and derivative of D can be accurately estimated, using the difference between neighbouring points. This results in a solvable 3×3 linear system [39, 7]. If $\hat{\mathbf{x}}$ is larger than 0.5 in any dimension, the extremum lies closer to a different point, so the point is changed and

the check repeated. The last \hat{x} offset, is added to its sample pixel location, which results in the estimate of extremum location.

This serves the dual purpose of removing low contrast pixels, because of their propensity for noise, and improving stability by removing edge responses. Full discussion of the theoretical underpinnings of these methods are in the literature already highlighted [7, 39].

Orientation Assignment

The central principle to Orientation Assignment is consistency across all keypoints based on local image properties. With this approach, the keypoint descriptors can be represented relative to orientation, thus becoming invariant to image rotation [39].

The keypoint scale is used to select the Gaussian image L . Matching the keypoint and the Gaussian as closely as possible leads to all computations having scale invariance. For each image sample $L(x, y)$ in any given scale, the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$, is precomputed using the difference between pixels in the following way:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}, \quad (3.7)$$

$$\theta(x, y) = \tan^{-1} \left((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y)) \right). \quad (3.8)$$

A histogram is created from the gradient orientations of the sample points in a region around the keypoint. This orientation histogram consists of 36 bins encompassing the full 360° range of orientations.

A Gaussian-weighted circular window and the keypoints' own gradient magnitude is used to weight the sample as it is added. The Gaussian window uses a σ equal to 1.5 times the scale of the keypoint. Within this scheme, any peaks in the orientation histogram correspond to local gradient dominant directions [39].

Any local peak within 80% of the highest peak is used to create another keypoint with that orientation. It follows that locations with peaks of similar magnitude can generate multiple keypoints, that would occur at the same location and scale but with different orientations [39]. These variances have a large effect on stability of the feature matching process.

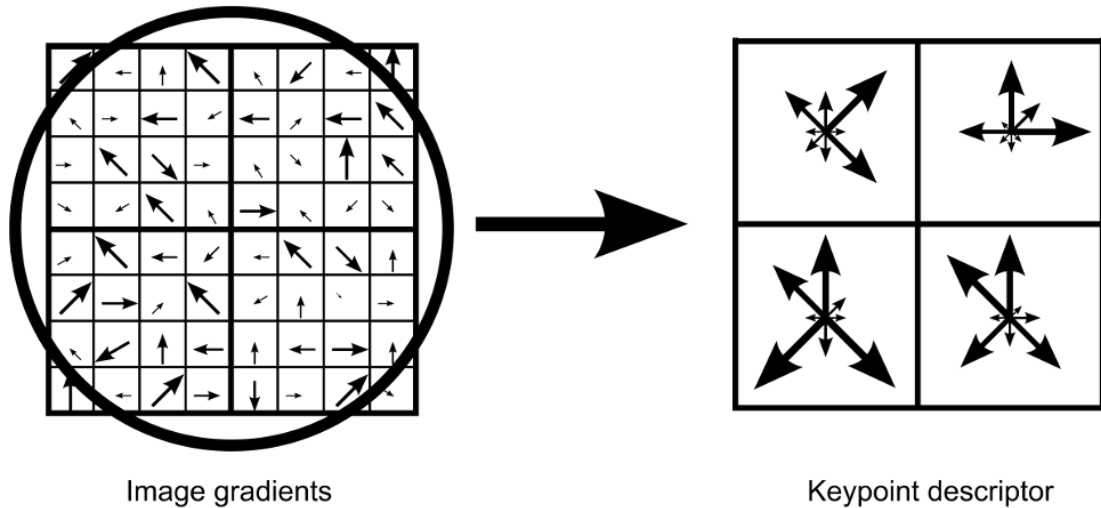


Figure 3.3: The Local Image Descriptor, as demonstrated by Lowe. Image gradients are aggregated to form a descriptor, and image gradient magnitudes are used to collectively weight the overall orientation for each keypoint descriptor.

The Local Image Descriptor

So far, each keypoint has been progressively further characterised via the application of scale, location and orientation. Practically, this characterisation is defined through the application of a localised planar (2D) coordinate system, the result of which, is the ability to describe the *local image region*. This gives the property of invariance to these parameters. So, the next step is the descriptor for the local image region.

The computation of the keypoint descriptor first requires the sampling of the gradient magnitudes and orientations around the keypoint location, utilising the keypoint scale as a factor in the level of Gaussian blur for the image. The coordinates of the descriptor and the gradient orientations are rotated with respect to the keypoint; this step provides *invariance to orientation*. A Gaussian function with a sigma equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point.

The keypoint descriptor is formed from a vector containing all values for all the orientation histograms. Lowe et al. [39] stipulates that best results are obtained using a 4×4 array of histograms with 8 orientation bins each, which expands to $4 \times 4 \times 8 = 128$ element *feature vector* for each keypoint. Lastly, the feature vector is normalised to unit length, which makes the vector invariant contrast change. Lowe et al. [39] states that any change in image contrast, in which a constant is added to each image pixel will multiply the image gradients by the

same constant, so the contrast change will be cancelled by vector normalisation, leading to *illumination invariance* for *linear* contrast changes.

This is a brief outline of the main stages/components of the SIFT algorithm. There are more components to the algorithm but these serve more ancillary roles, or apply a level of detail that is beyond the scope of this document. Full descriptions of all aspects of the algorithm design and implementation can be found the literature regarding SIFT [39, 40].

3.3 SfM: Bundle Adjustment and Reconstruction from Unorganised Datasets

The following subsection describes the theoretical underpinning for the Bundle Adjustment stage of the archetypal SfM algorithmic process. Like SIFT, there are several individual components that integrate and these will be technically explained but the discussion will be abridged. Full descriptions on the detail [and variations] of the process can be found in the literature [7, 6, 55, 62, 63, 22, 61, 10, 38].

As a brief initiatory description, Bundle Adjustment can broadly be termed as the simultaneous solution, defining the spatial coordinates that define scene/object geometry, relative motion and optical view characteristics, corresponding to the image projections of all points.

3.3.1 Pre-Bundle Adjustment: Image Matching

The following step to Feature Matching is *Image Matching*. This is the process of matching all images in the set, that can be matched. Practically this means finding all images that have the set of common spatial points. Commonalities will connect larger sets together, and the overall connected set will become a 3D model.

This requires the evaluation of every image against every other image in the set. The apparent quadratic nature of the matching process is mitigated by the practical need to only match a small number of neighbouring images, to obtain good camera position solutions. The following section on Image Matching and Bundle Adjustment draws heavily on the work of Brown [7, 6], Wu [62, 63] and Triggs [55] in explanation of this process.

Image Matching Mechanics

The mechanics of the Image Matching process are outlined here, as described in the literature [6]. A number m images, that have the largest number of feature matches to the current image, are used as potential image matches [7, 6]. Each camera is described using 7 parameters, grouped into three parts. These are: a rotation vector, $\Theta_i = [\theta_{i1} \ \theta_{i2} \ \theta_{i3}]$, translation vector, $\mathbf{t}_i = [t_{i1} \ t_{i2} \ t_{i3}]$ and focal length scalar, f_i . The calibration matrix is:

$$\mathbf{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.9)$$

The rotation matrix is described as:

$$\mathbf{R}_i = e^{[\theta_i]_{\times}} \quad (3.10)$$

where

$$[\theta_i]_{\times} = \begin{bmatrix} 0 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 0 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 0 \end{bmatrix}.$$

Each pairwise image match adds four camera parameter constraints, and three unknown structure parameters as follows: $\mathbf{X} = [X_1, \ X_2, \ X_3]$:

$$\tilde{\mathbf{u}}_i = \mathbf{K}_i \mathbf{X}_{ci} \quad (3.11)$$

$$\tilde{\mathbf{u}}_j = \mathbf{K}_j \mathbf{X}_{cj} \quad (3.12)$$

$$\mathbf{X}_{ci} = \mathbf{R}_i \mathbf{X} + \mathbf{t}_i \quad (3.13)$$

$$\mathbf{X}_{cj} = \mathbf{R}_j \mathbf{X} + \mathbf{t}_j, \quad (3.14)$$

where $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{u}}_j$ are the image position in camera i and j . The remaining constraint confirms the co-planar relationship between the two camera rays $\tilde{\mathbf{p}}_i$, $\tilde{\mathbf{p}}_j$ and the translation vector between camera centres \mathbf{t}_{ij} . The scalar triple product is thus zero:

$$\tilde{\mathbf{p}}_i^{\top} [\mathbf{t}_{ij}]_{\times} \tilde{\mathbf{p}}_j = 0. \quad (3.15)$$

In terms of camera parameters, it is found that:

$$\tilde{\mathbf{p}}_i = \mathbf{R}_i^{\top} \mathbf{K}_i^{-1} \tilde{\mathbf{u}}_i \quad (3.16)$$

$$\tilde{\mathbf{p}}_j = \mathbf{R}_j^{\top} \mathbf{K}_j^{-1} \tilde{\mathbf{u}}_j \quad (3.17)$$

$$\mathbf{t}_{ij} = \mathbf{R}_j^{\top} \mathbf{t}_j - \mathbf{R}_i^{\top} \mathbf{t}_i. \quad (3.18)$$

Substitution gives:

$$\tilde{\mathbf{u}}_i^T \mathbf{F}_{ij} \tilde{\mathbf{u}}_j = 0, \quad (3.19)$$

where

$$\mathbf{F}_{ij} = \mathbf{K}_i^{-T} \mathbf{R}_i [\mathbf{R}_j^T \mathbf{t}_j - \mathbf{R}_i^T \mathbf{t}_i] \times \mathbf{R}_j^T \mathbf{K}_j^{-1}. \quad (3.20)$$

This final expression is also known as the *epipolar constraint* [6]. Image Matching requires the accurate estimation of the fundamental matrix, \mathbf{F}_{ij} . RANSAC is employed to estimate \mathbf{F} and find the set of inlier points that share consistent epipolar geometry.

As mentioned above, 3D objects and scenes can then be identified and accurately extracted, as the connected components of image matches. As mentioned above, 3D objects and scenes can then be identified as the connected components of image matches.

3.3.2 Bundle Adjustment

The Bundle Adjustment step moves through the process of visual reconstruction, such that the optimal 3D structure and viewing parameters (camera position, orientation, calibration estimates) are extracted [55].

The process is named in reference to the concept of light rays leaving each 3D feature and converging on each camera centre, with respect to both feature and camera positions. Variables with respect to both the structure of the scene and the camera parameters are adjusted together as a single “bundle” [55].

Generally, bundle adjustment is tackled as non-linear least squares problem, with a cost function generally assumed to be of quadratic complexity, in both its feature re-projection errors and general robustness [55]. This approach is considered and described in more detail below.

Bundle Adjustment: Theoretical Underpinnings

As an optimisation problem, Bundle Adjustment is usually considered from a least squares perspective, with the error being defined as the squared L_2 norm of the difference between the observed feature location, and the projection of the corresponding point on the camera image plane [62]. This definition is important as it describes exactly the theoretical relationship by which 3D structure can be extracted from a plane representation.

If x is a vector of parameters and $f(x) = [f_1(x), \dots, f_k(x)]$ be the vector of reprojection errors for the 3D reconstruction, then the optimization problem to be solved is the non-linear least squares problem [62]:

$$x^* = \arg \min_x \sum_{i=1}^k \|f_i(x)\|^2. \quad (3.21)$$

The Levenberg-Marquardt (LM) algorithm is most commonly used for solving for bundle adjustment and other problems of a similar type. It works by solving a series of linear approximations of the original problem [62]. If $J(x)$ is the Jacobian of $f(x)$, then a linear least squares problem of the following form is solved:

$$\delta^* = \arg \min_{\delta} \|J(x)\delta + f(x)\|^2 + \lambda \|D(x)\delta\|^2. \quad (3.22)$$

This occurs in every iteration and updates $x \leftarrow x + \delta^*$ if $\|f(x + \delta^*)\| < \|f(x)\|$. The result, $D(x)$, is a non-negative diagonal matrix, usually the square root of the diagonal of the matrix $J(x)^T J(x)$, with λ being a non-negative parameter that dictates convergence behaviour. LM updates the value of λ at each step, based on how well the $J(x)$ approximates $f(x)$ [62]. Solving 3.22 is equivalent to solving the following normal equations:

$$(J^T J + \lambda D^T D)\delta = -J^T f. \quad (3.23)$$

This can be reduced to a structure called the *Augmented Hessian Matrix*:

$$H_{\lambda} = J^T J + \lambda D^T D. \quad (3.24)$$

The parameter vector is organized as $x = [x_c; x_p]$, where x_c and x_p are the parameter vectors for the camera and the point, respectively. The same convention is adopted for D , δ , and J , where the subscripts c and p indicate the camera and point parts. If the quantities are redefined [62]:

$$U = J_c^T J_c, \quad (3.25)$$

$$V = J_p^T J_p, \quad (3.26)$$

$$U_{\lambda} = U + \lambda D_c^T D_c, \quad (3.27)$$

$$V_{\lambda} = V + \lambda D_p^T D_p, \quad (3.28)$$

$$W = J_p^T J_p, \quad (3.29)$$

when 3.23 can be re-written as the following structured linear system [62]:

$$\begin{bmatrix} U_\lambda & W \\ W^\top & V_\lambda \end{bmatrix} \begin{bmatrix} \delta_c \\ \delta_p \end{bmatrix} = - \begin{bmatrix} J_c^\top f \\ J_p^\top f \end{bmatrix} \quad (3.30)$$

By applying Gaussian elimination to the point parameters, a system of only camera parameters is extracted [62]:

$$(U_\lambda - WV_\lambda^{-1}W^\top)\lambda_c = -J_c^\top f + WV_\lambda^{-1}J_p^\top f. \quad (3.31)$$

The matrix $S = U_\lambda - WV_\lambda^{-1}W^\top$ is known as the *Reduced Camera Matrix* (or *Schur Complement*). Given the solution to 3.31, δ_p , the point parameters are extracted through substitution [62]:

$$\delta_p = -V_\lambda^{-1}(J_p^\top f + W^\top \delta_c). \quad (3.32)$$

This concludes the general explanation of the Bundle Adjustment problem. This results in the sparse point matrix (or sparse point cloud), S , and the corresponding camera and point vectors. There are drawbacks to this method: scale of the results produced directly influence computational cost, and so prevent the application of this generalised method for very large problems [62].

Solutions to this exist, such as *Preconditioned Conjugate Gradients*, which, through the use of a matrix vector product structure, find the result without explicitly unpacking all the quantities involved. Further details on this solution are available in the literature [62] but are beyond the scope of this monograph.

3.4 SfM: Patch-based Multi-View Steropsis

It is now common for modern MVS algorithms to achieve a relative accuracy in the region of 1 mm for every 20 cm of object, and this from “low resolution 640×480 images. There are four main types of these algorithms:

1. Voxel Based,
2. Deformable Polygonal Meshes,
3. Multiple Depth Map Methods,
4. Patch-based Methods.



Figure 3.4: After densification, it can be noted that there is more definition than after bundle adjustment. However, clusters of noise are still present and require removal.

The focus here is on Patch-based methods. These represent surfaces by collections of small patches (referred to as *surfels*) [23]. This works well for visualisation via point based rendering.

In this case, for brevity, the Furukawa PMVS methodology is the only considered. In this approach, the core algorithm reconstructs a set of oriented points (or patches) covering the surface of a scene. What follows is an abridged description of the mechanisms of operation for this method, and the following section will draw heavily on the work of Furukawa and Ponce [23], in explanation.

3.4.1 The Patch Model

The first step in grasping the workings of this method is to understand the concept of the "Patch" and define exactly what it is. A patch p is a local rectangular tangent plane approximation of a surface, oriented such that one of its edges is parallel to the x-axis of the viewing camera. Its geometry is fully described by its centroid point $\mathbf{c}(p)$, unit normal vector $\mathbf{n}(p)$, which, points directly at the camera plane, and a *reference image* $R(p)$, in which p is visible (this is the viewing camera). The boundary of the rectangle is defined such that the smallest square in $R(p)$ is of size $\mu \times \mu$ [23].

3.4.2 Photometric Discrepancy Function

The photometric discrepancy function is the mechanism by which the spatial reference geometry of a patch is derived. Separate reference images of a patch evaluated against each other by the function, leading to a characterisation of the patch different to both. The scheme can be described by using $V(p)$ as the expression of the set of images in which p is visible. The discrepancy function is thus [23]:

$$g(p) = \frac{1}{|V(p)/R(p)|} \sum_{I \in V(p)/R(p)} h(p, I, R(p)), \quad (3.33)$$

where $h(p, I_1, I_2)$ is the pairwise photometric discrepancy between the images I_1 and I_2 . Practically, the computation of this quantity is done with the following steps [23]:

1. Overlaying a $\mu \times \mu$ grid on p ,
2. Using bilinear interpolation to sample pixel colours $q(p, I_i)$ of all grid points in each image I_i^2 ,
3. Computing $1 - |S_{cc}(q(p, I_1), q(p, I_2))|$, where S_{cc} is the cross correlation score between $q(p, I_1)$ and $q(p, I_2)$.

Key to remember, is that only images that have a photometric discrepancy score with $R(p)$ below a threshold α are used for evaluation. This leads to the following with respect to $V(p)$ and $g(p)$ [23]:

$$V^*(p) = \{I | I \in V(p), h(p, I, R(p)) \leq \alpha\}, \quad (3.34)$$

$$g^*(p) = \frac{1}{|V^*(p)/R(p)|} \sum_{I \in V^*(p)/R(p)} h(p, I, R(p)). \quad (3.35)$$

3.4.3 Image Model

The patch-based representation is highly flexible but there is a lack of connectivity information in this process, thus it is not readily possible to access or search for neighbouring patches. The Furukawa and Ponce method, combats this by maintaining of a record of the image projections of reconstructed patches, for each corresponding source image.

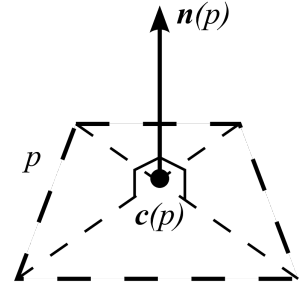


Figure 3.5: Representation of a PMVS patch, p , defined primarily by its normal, n , and its centroid point, c

Practically, this consists of associating groups of connected images, I_i together in aggregate grid structures of $\beta \times \beta$ pixels. These are known as *cells*, $C_i(x, y)$. Once this is done, each patch is projected onto each image until the matching cell is found. Each cell $C_i(x, y)$, retains a reference to the set of patches $Q_i(x, y)$ that project into it [23].

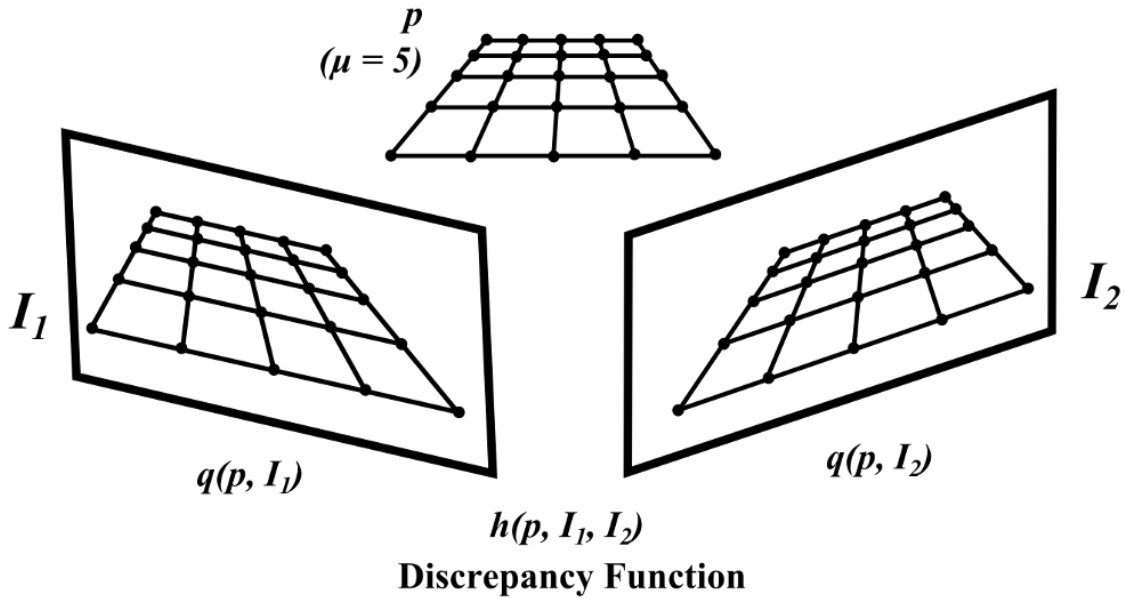


Figure 3.6: Visual description of the Photometric Discrepancy Function, as demonstrated by Furukawa and Ponce. The definitive geometry of a patch, p , is obtained through the evaluation of multiple reference images, $R(p)$, where p is visible.

3.4.4 Patch Reconstruction

The goal with the Furukawa method is to reconstruct at least one patch in every image cell. This process is broken down into three steps. Short descriptions of these follow.

Initial Feature Matching

The Initial feature matching procedure is where corner and blob features are detected via DoG and Harris filters, in a process of *image convolution*. Feature matching proper consists of the collecting all images within the set, with features of the same type, and that also lie close to the corresponding epipolar lines [23].

The spatial points of these images can then be associated. Points are considered in order of increasing distance from the camera optical origin. Then are categorised as potential patch centers, and a patch attempted to be generated from each.

Feature Matching

For this, consider an image I_i and the optical center of a corresponding camera, $O(I_i)$. For each feature f detected in I_i , in the other images, the set F of features f' , that are within 2 pixels of the nearest epipolar lines. From this, the spatial points associated with the pairs (f, f') are triangulated. They are then evaluated as potential patch centers, as they increase in distance from $O(I_i)$. The following procedure initialises a patch candidate p , with corresponding parameters $\mathbf{c}(p)$, $\mathbf{n}(p)$ and $R(p)$, with respect to a feature pair, (f, f') [23]:

$$\mathbf{c}(p) \leftarrow \{Triangulation(f, f')\}, \quad (3.36)$$

$$\mathbf{n}(p) \leftarrow \frac{\overrightarrow{\mathbf{c}(p)O(I_i)}}{|\overrightarrow{\mathbf{c}(p)O(I_i)}|}, \quad (3.37)$$

$$R(p) \leftarrow I_i. \quad (3.38)$$

To combat initial patch sparsity and the propensity for false positives in the first instance, it is assumed that the patch is visible in I_i , when the difference in angle between $\mathbf{n}(p)$ and $O(I_i)$ is below a threshold, ι [23]:

$$V(p) \leftarrow \{I|\mathbf{n}(p) \cdot \overrightarrow{\mathbf{c}(p)O(I)} / |\overrightarrow{\mathbf{c}(p)O(I)}| > \cos(\iota)\}. \quad (3.39)$$

3.4.5 Patch Optimisation

With the photometric discrepancy function $g^*(p)$, the patches with the smallest discrepancy scores can be identified and integrated. Each patch p is constructed in two distinct steps [23]:

1. initialisation of the corresponding parameters: $\mathbf{c}(p)$, $\mathbf{n}(p)$, $R(p)$, and the visible images, $V^*(p)$,
2. optimisation of the geometric component: $\mathbf{c}(p)$, $\mathbf{n}(p)$.

The geometric parameters are optimised through minimisation of $g^*(p)$ with respects to the corresponding parameters. If $\mathbf{c}(p)$ is constrained in the correct way, the degrees of freedom can be reduced to one and only depth will need to be solved for. Similarly, if $\mathbf{n}(p)$ is parametrised by Euler angles, the resultant problem only has three parameters.

3.4.6 Expansion

In this step, at least one patch is reconstructed in every cell, and the process is repeated through looking at the empty spaces between the patches in cells and going through the patch generation process there, after the evaluation of certain criteria.

In this step, at least one patch is reconstructed in every cell, with the aim being to repeat the process, generating new patches in empty spaces by identifying the set of neighbouring cells resident in each image, I_i [23]:

$$\mathbf{C}(p) = \{C_i(x', y') | p \in Q_i(x, y), |x - x'| + |y - y'| = 1\} \quad (3.40)$$

To generate a new patch p' , the expansion procedure for each image cell $C_i(x, y)$ in $\mathbf{C}(p)$ is as follows [23]:

- Initialise $\mathbf{n}(p')$, $R(p')$, $V(p')$ by the corresponding values of p ,
- Initialise $\mathbf{c}(p')$ as the point where the ray through the centre of $C_i(x, y)$ intersect with the plane of p ,
- Proceed to feature matching and optimisation procedure, producing $V^*(p')$, and refining $\mathbf{c}(p')$ and $\mathbf{c}(p')$,
- Add $V(p')$ to a set of images in which the patch should be visible according to a depth-map test,
- The expansion is accepted if $|V^*(p')| \geq \gamma$, where γ is a threshold number of images with low photometric discrepancy values.

3.4.7 Filtering

Three filters are applied to exclude inconsistent patches. The filters maintain visibility consistency and inequality filtering, and, the reconstructed patches form an oriented point, or surfel model [23].

1. If p and p' are not direct neighbours but are both within the same cell, p is filtered out if the following condition is met:

$$|V^*(p)|(1 - g^*(p)) < \sum_{p_i \in U(p)} 1 - g^*(p_i), \quad (3.41)$$

where $U(p)$ denotes, those patches p' , that are not direct neighbours with p .

2. The second filter excludes p , if the number of images in $V^*(p)$ where p is visible, is less than γ , according to a depth-map test,
3. The third filter captures all patches in the current and adjacent cells in all $V(p)$, and determines the proportion that qualify as neighbours of p , according to the following scheme:

$$|(\mathbf{c}(p) - \mathbf{c}(p')) \cdot \mathbf{n}(p)| + |(\mathbf{c}(p) - \mathbf{c}(p')) \cdot \mathbf{n}(p')| < 2\rho_1, \quad (3.42)$$

If the output is lower than 25%, p is also excluded.

This concludes the section densification of sparse point clouds, via the Furukawa and Ponce PMVS algorithm. Greater detail is available in the literature [23], where full discussion on all areas covered here are explored.

3.5 Meshing/Tessellation: Poisson Surfacing

This section borrows heavily from the literature surface tessellation and PSR, and, in particular the work of Sorkine [53], Furukawa [23], Kazhdan [33, 34] and Calacki [8].

3.5.1 Surface Reconstruction: Combinatorial Techniques

The problem of surface reconstruction in three dimensions is a well studied problem, in many areas. Several approaches exist, using various principles. These include Delaunay Triangulation, Alpha Shapes and Voronoi Diagrams.

Generally, each of these combinatorial methods, create triangular meshes that interpolate, at least, some of the points. This mitigates inherent problems with many datasets, such as non-uniform sampling and sampling accuracy/inconsistency, which lead directly to point and normal noise effects [33, 34]. Many techniques have the drawback that in the presence of noisy data, the resulting surface is often jagged, and so needs to be further processed to achieve smoothing [33, 34].

3.5.2 The Poisson Technique

As well as finding applications in many other unrelated areas, Poisson's equation is very effective at reconstructing a smooth surface, based on a space containing a number of points \mathbf{p}_i , where each point also has associated with it a local surface normal \mathbf{n}_i .

As mentioned above, although there are various techniques that can be applied, there are several unique advantages to formulating surface reconstruction as a Poisson problem [33, 34].

Many of the techniques used in this area segment the data into regions for local fitting and then recombine the approximations, to make the whole. This methodology leads to problems with scaling, smoothing and artefacts being introduced into the data. In contrast, the Poisson solution treats the point cloud globally with no dependence on heuristic partitioning or blending functions [33, 34]. This leads to the creation of very smooth surfaces, where noisy data is handled in a more robust fashion [33, 34]. Incorporated is a hierarchy of locally supported functions, allowing the solution to reduce to a sparse linear system. This makes Poisson system highly resilient to ill-fitting data, also.

Solving the Implicit Function: χ

PSR reconstructs the implicit function χ , whose value is zero at the points \mathbf{p}_i and whose gradient at the points \mathbf{p}_i equals the inward facing normal vectors \mathbf{n}_i .

The set of $(\mathbf{p}_i, \mathbf{n}_i)$, is thus a sampling of a continuous vector field \vec{V} on the volume V . The gradient of the function χ is described by the best approximation to the vector field \vec{V} :

$$\min_{\chi} \|\nabla\chi - \vec{V}\|. \quad (3.43)$$

Application of the divergence operator transforms the variational problem into a Poisson problem: the scalar function χ whose Laplacian (divergence of gradient) is equal to the vector field divergence [33]:

$$\Delta\chi \equiv \nabla \cdot \nabla\chi = \nabla \cdot \vec{V}. \quad (3.44)$$

3.5.3 Octrees

Going hand in hand with the PSR as a method of detail control is the *Octree*. Practically within PSR, this finds expression as a continual division of space where structure exists, while

ignoring areas of no structure. This effect of this is best illustrated with *primal graphs* [8].

Defining the Octree

An Octree is a cell based, *3D tree data structure, recursive subdivision algorithm*, in which each internal node has exactly eight child nodes. It is adaptively constructed, based on the location of the data points on which it is operating [8].

A cell is split if the number of points contained is larger than a pre-specified value, and, the cell depth does not exceed another pre-specified value [8]. The data set is recursively partitioned into subsets of points $D\alpha$, associated with cells $C\alpha$. For Octrees, the multi-indices $\alpha = (L, i, j, k)$ must be augmented with a level (or depth) value L , both for cells and vertices, where $0 \leq L \leq LMAX$ [8]. In many PSR algorithms and implementations, the Octree serves as one of the main methods of dictating the fidelity, and depth of detail of reconstruction [8].

3.6 Extracting Fracture Orientations

The last consideration in this background section is the algorithmic process of determining the fracture orientation, Γ , of surfaces. What follows is an explanation of how accurate orientations can be automatically derived, using known quantities and the interpretation of these quantities into geological measurements, namely Strike(Γ_ζ), Dip Angle(Γ_α) and Dip Direction(Γ_δ).

Fracture Orientation refers to the spatial attitude of jointing or fracturing in geological rock units. This can be described by Γ_α and $\Gamma_\delta/\Gamma_\zeta$ of the plane. Key to understanding these relationships is the plane equation. This is generally defined by the relationship between a point, p , that forms the centroid of a plane and its normal vector, n , and has the following form:

$$n_{(a)} \times p_{(x)} + n_{(b)} \times p_{(y)} + n_{(c)} \times p_{(z)} + d = 0, \quad (3.45)$$

where a, b, c are the indices of the normal vector of the plane, and d is a constant determined by the vector dot product:

$$-\vec{p}_{(x,y,z)} \cdot \vec{n}_{(a,b,c)}. \quad (3.46)$$

What follows is a brief description of how to derive the three Fracture Orientation measurements, Γ_ζ , Γ_α and Γ_δ .

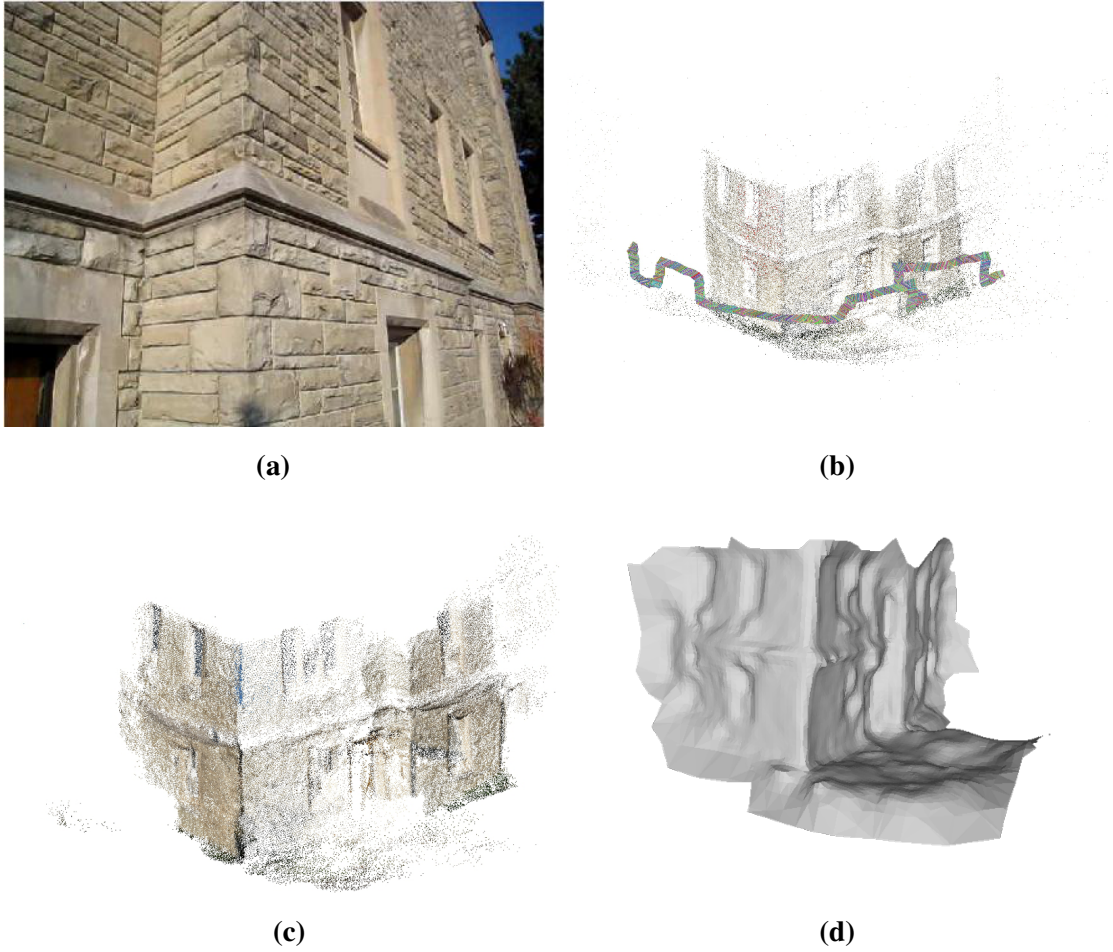


Figure 3.7: The Software Toolchain comprises four stages, consisting of several separate applications used to produce a Lidar-like point cloud. (a) Stage 0 and Stage 1, resulting in image acquisition; (b) Stage 2, resulting in the sparse point cloud; (c) The densification process, resulting in a more clearly differentiated point cloud; (d) Shows PSR meshing process, that produces a defined surface for further processing.

3.6.1 Dip Angle, Γ_α

Determining Orientation parameters is reasonably straightforward, if the surface plane is known. Γ_α is defined as the angle between the best-fit (or ideal) plane of the fracture surface and the [ideal] horizontal plane (90° facing azimuth). The Cartesian X-Y plane can be taken as the horizontal plane. The angle φ between the two planes can be determined by the dot product of their normal vectors, \mathbf{n} and \mathbf{n}_h [19]:

$$\cos \varphi = \frac{|\mathbf{n}_h \cdot \mathbf{n}|}{|\mathbf{n}_h| |\mathbf{n}|} = \left| \frac{c}{\sqrt{a^2 + b^2 + c^2}} \right|. \quad (3.47)$$

Γ_α is then resultant from [19]:

$$\arccos \left| \frac{c}{\sqrt{a^2 + b^2 + c^2}} \right|. \quad (3.48)$$

3.6.2 Dip Direction, Γ_δ , and Strike, Γ_ζ

The Strike of a surface is simply described as one of the two directions of the intersection between the fracture plane and the horizontal plane. The relationship between Γ_ζ and Γ_δ is generally defined with the right hand rule. That is, if standing along the intersected line and the rock fracture plane is dipping to the right-hand side, Γ_ζ is facing ahead. This makes the calculation $\Gamma_\zeta = \Gamma_\delta - 90^\circ$ [19]. In this coordinate system, the x-axis is aims in the direction

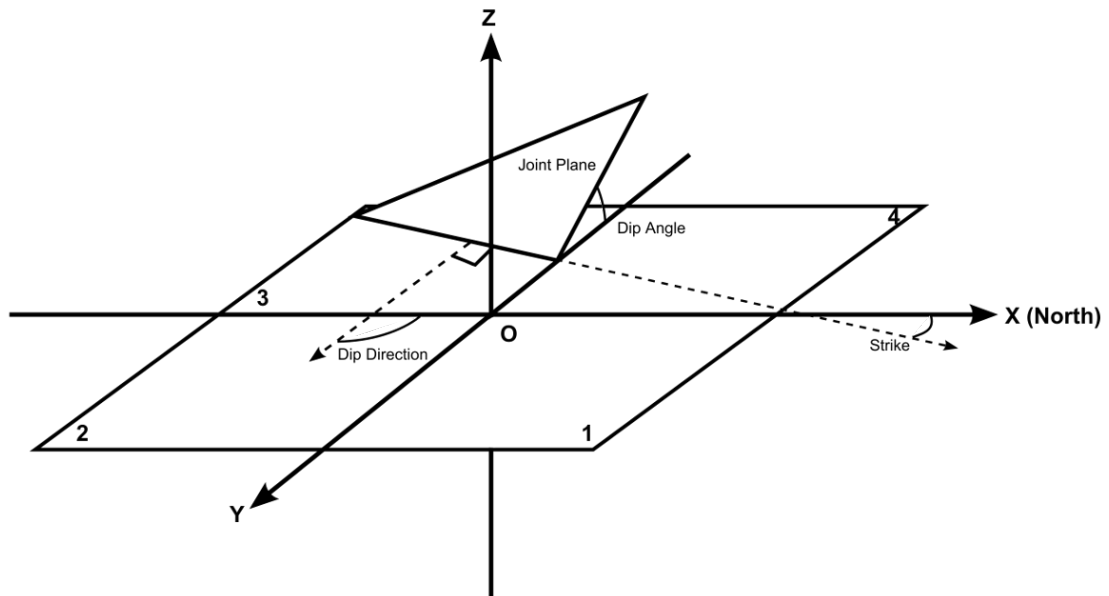


Figure 3.8: Visual representation of the geometric identities that are described using the fracture orientations: Γ_ζ , Γ_α and Γ_δ .

of **North**. The Γ_δ of the fracture plane is actually the azimuth of the normal vector of the fracture plane [19]. The angle, κ between the horizontal component of the normal vector of the fracture plane and the positive x-axis is [19]:

$$\cos(\kappa) = \frac{a}{\sqrt{a^2 + b^2}}. \quad (3.49)$$

This equation cannot directly determine Γ_δ for a couple of reasons. Firstly, the Γ_δ is in the range of $0-360^\circ$ but the above equation only has an angular range of $0-180^\circ$. Secondly, the

normal vector can face either upward or downward and in geology, the upward facing vector is generally what is accepted.

The next step is to use the directional cosines of the normal vector, \mathbf{n} , of the candidate plane. These can be determined via the following pattern [19]:

$$\left\{ \begin{array}{l} \cos(\alpha) = \frac{a}{\pm \sqrt{a^2+b^2+c^2}} \\ \cos(\beta) = \frac{b}{\pm \sqrt{a^2+b^2+c^2}} \\ \cos(\gamma) = \frac{c}{\pm \sqrt{a^2+b^2+c^2}} \end{array} \right\}. \quad (3.50)$$

α is the angle between the positive x-axis and \mathbf{n} , β is the angle between the positive y-axis and \mathbf{n} and γ is the angle between the positive z-axis and \mathbf{n} . If the plane equation constant $d < 0$, $\pm \sqrt{a^2 + b^2 + c^2}$ is positive. If $d \geq 0$, $\pm \sqrt{a^2 + b^2 + c^2}$ is negative. This leaves only the final determination of Γ_δ [19]:

1. $\Gamma_\delta = \arccos(\kappa)$, if $\cos(\gamma) \geq 0$ and $\cos(\beta) \geq 0$,
2. $\Gamma_\delta = 360 - \arccos(\kappa)$, if $\cos(\gamma) \geq 0$ and $\cos(\beta) < 0$,
3. $\Gamma_\delta = 180 + \arccos(\kappa)$, if $\cos(\gamma) < 0$ and $\cos(\beta) \geq 0$,
4. $\Gamma_\delta = 180 - \arccos(\kappa)$, if $\cos(\gamma) < 0$ and $\cos(\beta) < 0$.

3.6.3 The Discontinuity Problem

This orientation extraction method [19] has a systematic discontinuity in the calculation, where the $\cos(\gamma)$ changes from a positive value to a negative value.

This comes about where Γ_α of surfaces measure very close to 90° . In these cases the value of $\cos(\gamma)$ is negligible but, its sign causes a large change in orientation by applying the incorrect quantity to $\arccos(\kappa)$. Γ_δ (which can be thought of as the surface normal to the Γ_ζ direction) can therefore be translated incorrectly and give the wrong answer. By way of an example consider the following situation, involving a point \vec{p} and its associated normal \vec{n} . Following through the plane equation can be derived as:

$$n_a p_x + n_b p_y + n_c p_z + d = 0, \quad (3.51)$$

where

$$\vec{n} = \begin{pmatrix} -0.8931 \\ -0.4485 \\ -0.0348 \end{pmatrix} \quad (3.52)$$

and

$$\vec{p} = (28.3894, 1.5556, -2.6852). \quad (3.53)$$

Resulting in the following for $\cos(\kappa)$

$$\cos(\kappa) = \frac{a}{\sqrt{a^2 + b^2}}, \quad (3.54)$$

$$\cos(\kappa) = -0.8936, \quad (3.55)$$

$$\arccos(\kappa) = 153^\circ. \quad (3.56)$$

The constant d is the dot product of \vec{p} and \vec{n} :

$$d = -\vec{p} \cdot \vec{n}, \quad (3.57)$$

$$d = 25.96. \quad (3.58)$$

This quantity is positive, thus $\sqrt{(a^2 + b^2 + c^2)}$ is negative, leading to:

$$\cos(\beta) = \frac{b}{-\sqrt{a^2 + b^2 + c^2}}, \quad (3.59)$$

$$\cos(\beta) = 0.4485 \quad (3.60)$$

and

$$\cos(\gamma) = \frac{c}{-\sqrt{a^2 + b^2 + c^2}}, \quad (3.61)$$

$$\cos(\gamma) = 0.0348. \quad (3.62)$$

In this case, $\cos(\beta)$ and $\cos(\gamma)$ are positive, so the first of the above statements is used, leading to the following Γ_δ :

$$\Gamma_\delta = \arccos(\kappa), \quad (3.63)$$

$$\Gamma_\delta = 153^\circ. \quad (3.64)$$

Now, if it is assumed that there are minor inconsistencies between ground-truth and measured surfaces at the vertical, this can be simulated by changing the sign of the third normal element.

Now it is -0.0348. Following through again, now Γ_δ is determined by the third statement rather than the first:

$$\Gamma_\delta = 180 + \arccos(\kappa), \quad (3.65)$$

$$\Gamma_\delta = 333^\circ. \quad (3.66)$$

This example demonstrates how the orientation of one component of the normal vector can result in a significant difference in Γ_δ from minor change between ground-truth and measured quantities.

As a solution to this problem, a method has been implemented to compare the difference between ground-truth and measured normal vectors. This relationship less structurally complex, as there is no need to evaluate three separate quantities between ground-truth and reconstruction to understand each region. Also, importantly, this method can be evaluated uniformly reliable way, as it is a smooth, mathematically continuous function. This is fully discussed in Section 5.8

Chapter 4

Contributions

This chapter will outline the contributions made by the author in this monograph. There are, in general, three stages to this Software Toolchain: Data Preprocessing, Data Processing and Data Output/Orientation Measurement. Each of these stages consist of various algorithms and subroutines, that together, allow the extraction of large scale structure from a Software Generated Point Cloud (SGPC).

4.1 Overview Discussions

This chapter is organised as follows: a brief overview is given of the complete Toolchain, individual summaries of the Toolchain components, their functions and the contributions made within each, as applicable. The contributions made by the author are listed directly below for reference:

- Developed a high density sample rate strategy for the collection SfM data, too maximise resultant point cloud fidelity
- Developed the Software Toolchain as a whole by:
 - Integrating SfM and PSR software to produce a surface representation of Lidar type
 - Integrating a Mahalanobis distance methodology for anisotropic noise processing
 - Applied a voxel-fit methodology previously used on high density Lidar point cloud data

- Detected large scale regions in point cloud topology using a k NN Region Growing methodology
- Created an accurate bounded planar representation of regions using a localised coordinate reference methodology and a φ Golden Section Search to find the optimum perimeter for each
- Tested accuracy of reconstruction using a “minimum of normals” method, as an alternative to the standard direction cosines method for obtaining fracture orientations

This section describes the Toolchain as a whole, that being the main contribution of this thesis. Figure 4.1 provides a high level view of the processing stages of the Toolchain. The following subsections will describe each algorithm of the Toolchain, provide motivations for inclusion, development and implementation choices.

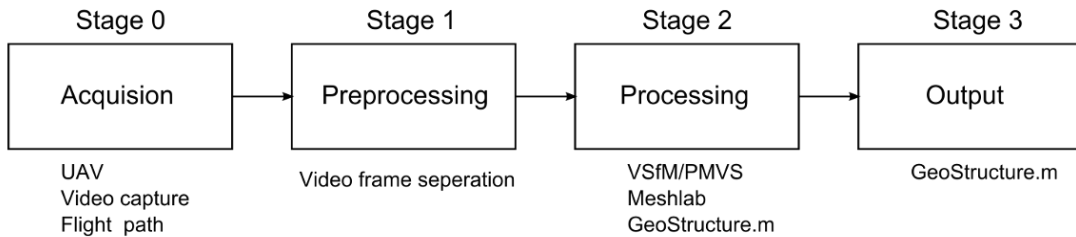


Figure 4.1: Overall work flow of the proposed system

The following section briefly describes the role that each component of the Toolchain has. The stages of processing are as follows:

4.2 Stage 0: Data Acquisition

While this is not strictly part of the Toolchain itself, there are some considerations in the data acquisition stage. As mentioned in the Background and Review sections, in order to maximise fidelity in the reconstruction of the scene, the number of unique views and angles of the scene are of prime importance.

To maximise this, while minimising other considerations in data acquisition process, it was determined that using video as the capture method [44], would enable a simpler acquisition strategy, while ensuring a much denser spread of views and angles.

4.2.1 Flight Path

Bemis et al. [3] cites good results generally coming from images with 50-60% overlap. Bemis [3] and James [31, 32] both indicate the value of oblique and converging images, respectively but in this small scale case, it is not required, as the splitting of frames should image overlap and obliquity are present in the image set.

This leaves only the Flight Path to strategise for and a uniform, recurrent path was chose to, again, maximise image overlap and obliquity characteristics of the set. This path is shown in figure 4.2, which illustrates how the bundle adjustment process also reconstructs the positions of the views used in the reconstruction. The flight path of the camera used is visible and highlighted in this example. The high density of view samples is also apparent, being as it appears as one continuous line.

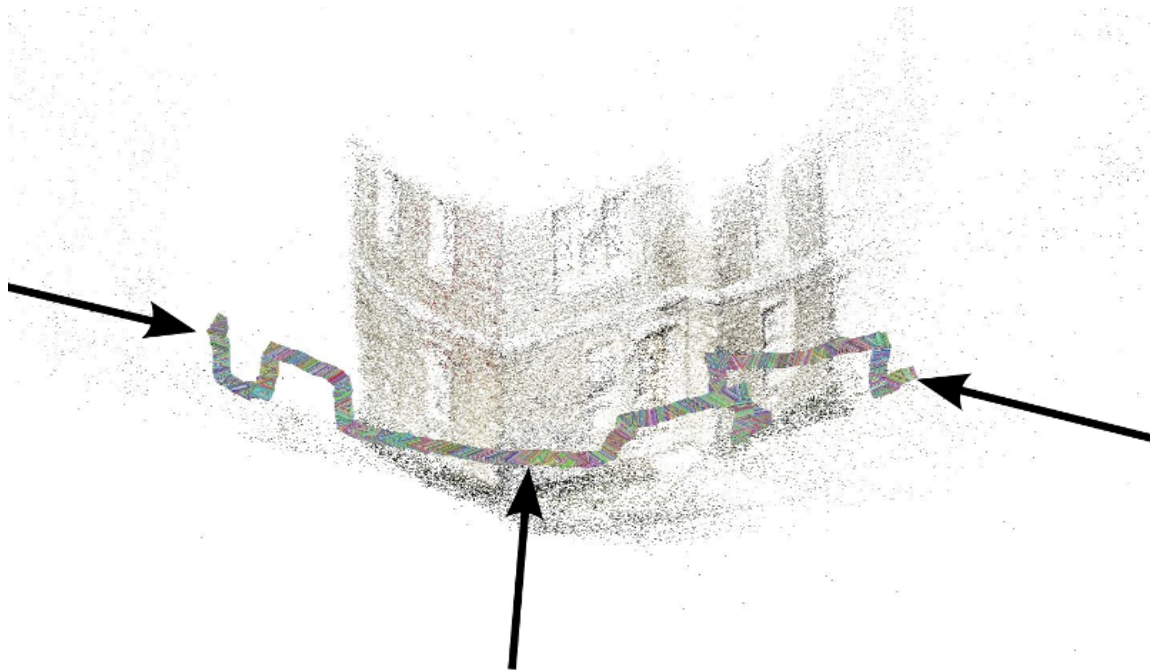


Figure 4.2: The flight path of the camera used is visible and highlighted in this example. The high density of views appear as one continuous line.

4.3 Stage 1: Preprocessing

As stated above, the data is acquired in video format, however unlike McLeod et al. [44], or Frahm [22], the Toolchain will use image-based SfM software (in this case VisualSfM [62, 63]), rather than attempting to implement a similar algorithm for video. The VSfM software has some requirements (such as only processing images in .jpeg format), and this requires some preprocessing in order to end up with a format usable in the next stage.

4.3.1 Frame Separation

The result of this first stage is an output movie file of the path of the camera. A subroutine was developed and used to perform the Preprocessing step. This was achieved via an algorithm written in Matlab. This program uses the `VideoReader.m` function to read the input file into a Matlab object, where the frames can be parsed and output to a specified folder, one at a time. This produces a dataset of images that can be input into the VSfM software.

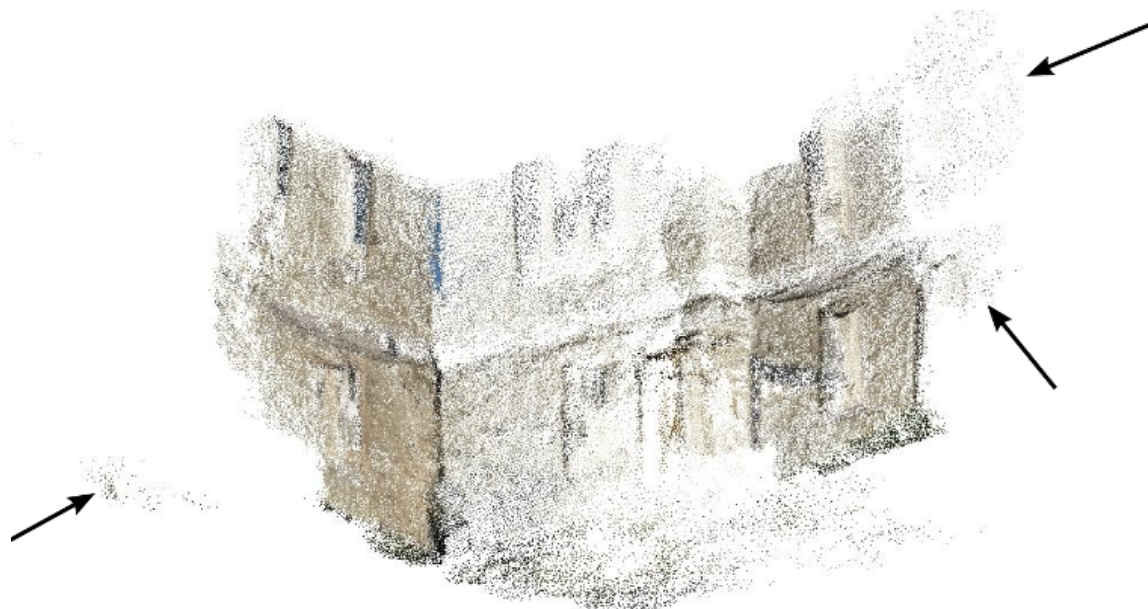


Figure 4.3: At completion of the bundle adjustment process, densification is very similar to anisotropic noise and is treated as such later in the cycle.

4.4 Stage 2: Initial Point Cloud generation (VSfM)

The next data processing stage is the SfM treatment using the VSfM software package [62, 63]. This stage of the Toolchain begins with the preprocessed image set, and ends with the sparse point reconstruction of the scene. The individual steps involved in this process encompass both SIFT processing and image matching on each image, block bundle adjustment and sparse point cloud generation. All of these are discussed in the Background and Review chapters.

4.5 Stage 2: Point Cloud Densification (PMVS plugin)

The densification stage take the sparse cloud and processes it to a “dense point cloud”. It uses tessellated patch identification and expansion techniques, to identify and interpolate structure in the sparse representation.

The densification process has the effect of selecting out a lot of noise and residual errors present in the sparse representation. However because the densification is purely a software-based process, it can also result in the addition of incorrect particles and noisy elements, as illustrated in figure 4.3. From this example, it can be noted that although there is far greater definition than at completion of the bundle adjustment process, densification still adds many clusters of the unwanted points(noise). These have qualities very similar to anisotropic noise and will be treated in a similar fashion later in the processing cycle. This is implemented PMVS plugin extension to the VSfM software and is discussed fully in the Background and Review chapters.

4.6 Stage 2: Surface Tessellation (Meshlab)

One problem with SGPC generation is the inevitable addition of noise and other errors through the algorithmic process. With the intention of processing the point cloud using surface detection and segmentation, into planes that can be measured, erroneous points could have a compounding effect in later stages, drastically affecting results.

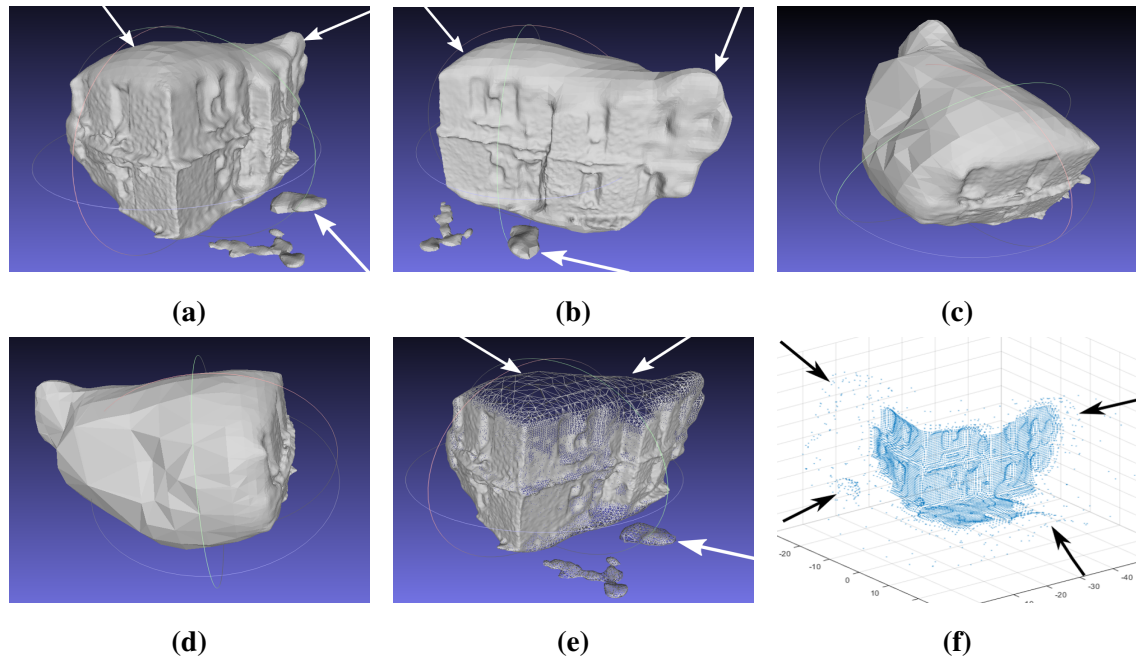


Figure 4.4: This above series of images indicates where every vertex in the bulbous area, results in an unwanted point, resulting in an anisotropic-type noise, fully visible in (f).

4.6.1 Utilising PSR

Here, the Poisson vector field equation is utilised to tessellate the dense point cloud into a solid surface. As stated above the PSR method has some uniquely useful characteristics: the property of treating the whole point cloud simultaneously [33, 34, 8], which leads to smoother, better fitted and more stable surfacing results [34]. The resolution of the mesh model is adaptive, meaning that the denser the points, the finer the tessellation. This results in a high fidelity and leads to a surface that is much closer in structure to the Lidar point clouds used in Gigli et al. [24]. The particulars of the PSR technique is discussed in the Background and Review sections.

4.7 Stage 2: GeoStructure.m

The majority share of the processing is done by the GeoStructure Processing Pipeline, and this constitutes the main contribution of this monograph. GeoStructure is a collection of algorithms and functions that process the point cloud through various stages and extract the fracture orientation measurements.

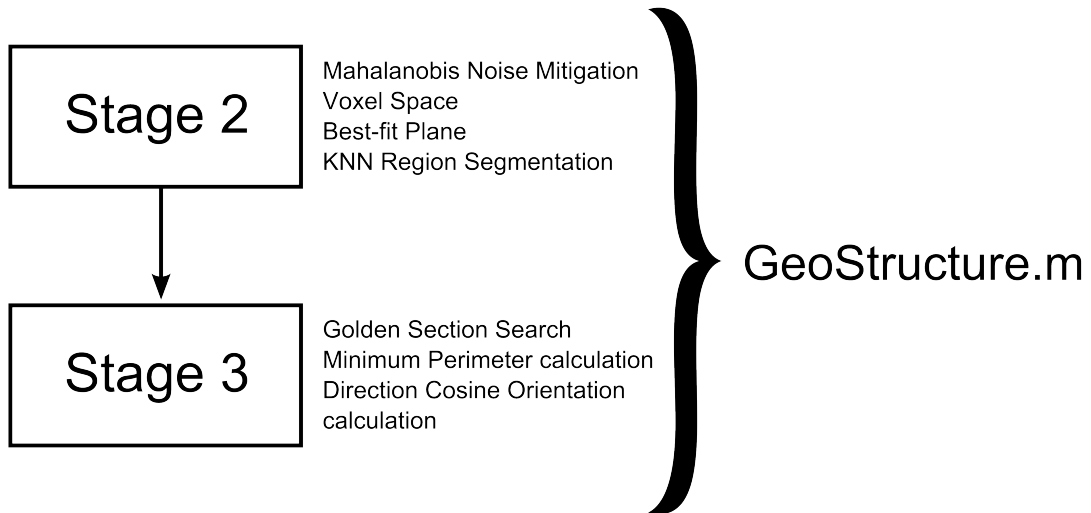


Figure 4.5: Outline of functions handled explicitly by the GeoStructure.m pipeline

The pipeline concept is based on the work of several researchers in geology, software, and, image processing and computer vision [24, 48, 23, 63, 52, 59, 3]. In particular, Gigli et al. [24] developed a software package to obtain fracture orientations from Lidar point clouds, which served as a conceptual template, although none of the actual functions or the development process is shared. GeoStructure is run after the PSR (using the Meshlab application) and the stages the software algorithm are as follows:

1. Read point cloud file,
2. Create point cloud volume [point space] of uniform dimension that can be volumetrically searched,
3. Search the volume, one proportionally sized voxel at a time,
4. Evaluate each voxel for the presence of points,
5. If points are present, attempt to fit a best-fit plane to the volume,
6. Segment to the point cloud by grouping planes into macroscale surface regions and differentiate,
7. Aggregate planes of same region into a singular plane approximating the overall best fitting orientation,
8. Extract orientation measurements from regions.

The next section provides a detailed description of the more significant stages of the GeoStructure processing pipeline.

4.7.1 Mahalanobis Noise Mitigation

One of the draw backs of the PSR tessellation method is the creation of erroneous extra tessellation, where the detail of the point cloud ends. These can either connect back on themselves, forming a bulbous undefined area, or flatten and extend out to to the maximum possible area of the reconstruction.

Introducing Anisotropic Noise

The PSR will join disconnected ends of the structure together and the mechanism by which this occurs can lead to the bulbous mesh structures (pictured in figure 4.4), characterised by larger tessellations, where the points become sparse. When reduced to a point cloud, the vertices of these larger triangles are rendered as a generally anisotropic-type of sparse noise. This is illustrated here in (a) - (d). (e) shows a wireframe representation of this tessellation example, where every vertex in the bulbous areas of non-structure, results in an unwanted point. This will have a general effect very similar to anisotropic noise. (f) illustrates such an effect.

Work by Brophy [5] into the extraction of structure from point clouds with high density noise, demonstrated the effectiveness of a Mahalanobis distance metric for removing extraneous points.

The Mahalanobis distance operates in the following way:

$$d_{\Sigma}(x, x_i) = ((x - x_i)^T H^{-1} (x - x_i))^{1/2}, \quad (4.1)$$

where the covariance matrix

$$H = DD^T, \quad (4.2)$$

is constructed using the following:

$$D = (x_1 - x, x_2 - x, \dots, x_n - x). \quad (4.3)$$

As a measure of the distance between a point and the distribution of the set along each principle component axis, it is very effective, and, has the properties of being scale invariant, unitless and respecting correlations in the set [5, 64, 52].

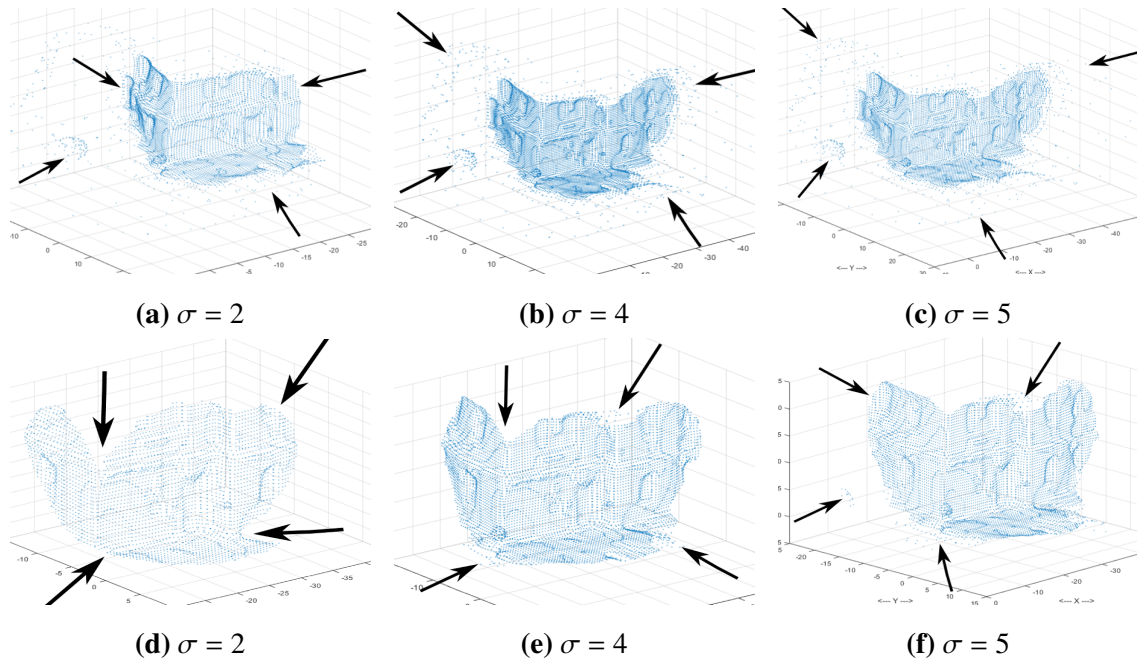


Figure 4.6: The above reconstructions compare the effect of a standard Euclidian distance method against a Mahalanobis method. The Euclidean method (a, b, c) shows significant noise with some legitimate structure no longer present. The Mahalanobis method (d, e, f) demonstrates significantly better behaviour at all σ distances.

Understanding the Effectiveness of the Metric

In a comparison of the removal of noisy points using a standard least squares method against using the Mahalanobis distance method, the Mahalanobis distance ratio of signal points removed to noise points is much better. With a least squares method of filtering, as the higher distance values are used, voids start to appear in the main structure, because this standard method has the effect of removing *all* points a certain distance from the centroid, which leads to problems where a non uniform structure is the subject.

The Mahalanobis metric recognises individual areas of density and correlation, and recognises those as part of the overall structure. The effect is that the metric is far more likely to recognise particular points as part of a structure (like a wall, for instance) and not remove those, and so, handles much higher distance values while maintaining the structures integrity.

Fig 4.6 illustrates this by showing how the PSR tessellation process can leave noise artefacts in the resulting reconstruction. The above reconstructions compare the effect of a standard Euclidian distance method (a, b, c) against a vanilla Mahalanobis method (d, e, f). The Euclidean method shows large numbers of noise clusters at a distance of 4 & 5 standard deviations (σ) from the mean, while at $\sigma = 2$, many noise clusters still occur, with voids present, over-

all structural point density visibly down and some legitimate structure no longer present. In contrast, the Mahalanobis distance demonstrates significantly better behaviour at all σ values. Noise clusters are greatly reduced at $\sigma = 4$ & 5, although some minor structural erosion has occurred. At $\sigma = 2$ there is more significant erosion but all noise has been filtered.

As mentioned above, this step is important because errant points will generate errant planes, and these will skew the orientation of region planes, later on in the process.

4.8 Stage 2: The Voxel-fit Process (Matlab)

As demonstrated in Gigli et al. and a multitude of other research [26, 58, 60, 44, 23, 8, 33, 34, 41, 5, 51], it is advisable to segment the point cloud into voxel grid, to extract structure. That process is detailed below.

4.8.1 Creating the Point Space

The *point space* is the total volume in which the point cloud is situated. As the following stage will require the division of this space into equally sized voxels [24] proportional to the overall size of the point space, the entire space is modified to be of equal length in all three directions, extended to the size of the longest dimension. This equalises all volumetric quantities and ensures that the point space consists of a whole number of voxels, and, that proportionality is preserved.

4.8.2 Searching the Point Space

A search cube (the *search space*), is then moved sequentially through each voxel in the grid. Ensuring the point space is equal length in all directions means that the size of the search space can be modified by dimensionless, unitless factoring value, that from this point forward, will be denoted by the letter ζ . Thus, a value of 0.5 for ζ , would create two search spaces in the point space, therefore producing two planes for the entire space; 0.25 would produce four search spaces, and so on. This removes the need for any arbitrary volume sizing values, while also allowing a good way of investigating the effect of plane size on the overall algorithm.

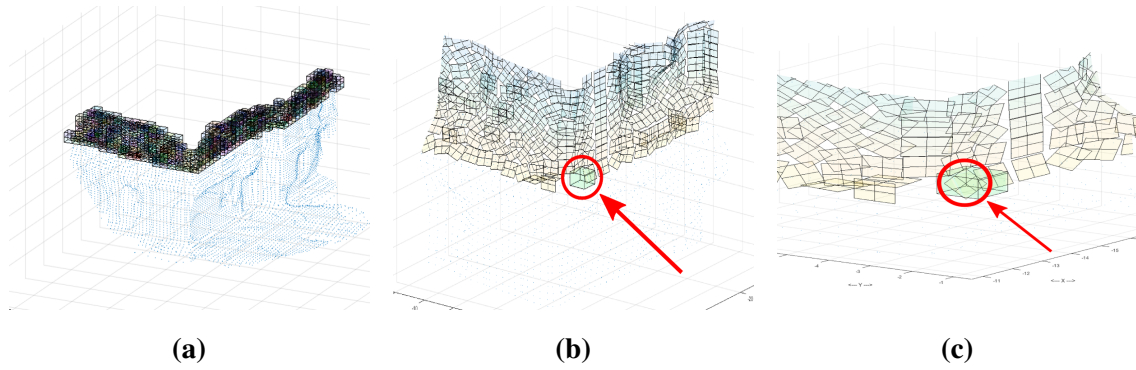


Figure 4.7: The voxel fitting process consists of the following stages: (a) division into equal sized volumes; (b) progression to next volume; (c) plane of current voxel-fit.

4.8.3 Applying the Voxel-fit

The voxel fitting process consists of the point space being subdivided into equal sized volumes and each of those being treated. This is visually described in fig. 4.7. (a) shows the division into equal sized volumes; (b) with only the current voxel is highlighted, and resultant voxel planes from previous fittings visible; (c) shows the fitting of the current voxel, with the resulting plane highlighted.

In each voxel, a set of operations is performed that results in the application of a least squares plane fit, or *voxel-fit*, to that voxel. The plane of the voxel is extracted by taking the mean of the points present in that voxel. All the points in the volume are then minimised with respect to the newly defined plane point, producing a modified point sample.

The principle directions of this sample set is then computed through an eigenvector decomposition. The first column of the resulting matrix produces the normal vector for the plane point, while the following two form the orthonormal base for the plane, allowing the construction of a voxel-fit for the voxel. Once this is complete, the search space moves on to the next voxel for fitting until all voxels have been treated in this fashion.

Evaluating the Search Space

Each voxel, v , is a volume of space bounded by the Real numbers in an orthogonal 3-directional co-ordinate system, \mathbb{R}^3 . In this case, v exists as part of a grid of identical voxels constituting the point space, pS , where each voxel, v_{pS} , in turn, is designated a search space, v_{SS} .

v_{SS} is evaluated for each point, p_i , of the overall point cloud set, p , where p is of size $n \times 3$.

If points are found, the centroid point C and it's normal vector $\vec{n}(C)$ are calculated. Firstly, in each voxel, v , the mean of the points, \bar{p}_i , belongs to the best-fit (voxel-fit) plane, V , and is the centroid of the search space and the plane:

$$C = \{\bar{p}_i | p_i \in v_{ss}\}, \quad (4.4)$$

meaning that,

$$\bar{p}_i = c(v_{ss}) = c(V). \quad (4.5)$$

The least squares distance between each of the points in the space and the centroid point is calculated:

$$\alpha x + \beta y + \gamma z + d = 0, \quad (4.6)$$

$$d = -\alpha c_x - \beta c_y - \gamma c_z, \quad (4.7)$$

which results in the *nominal plane equation*:

$$\alpha(x - c_x) + \beta(y - c_y) + \gamma(z - c_z) = 0. \quad (4.8)$$

It is important to remember that the points of interest here, p_i , are not necessarily on the plane; as C is an *ideal point*, it creates a *ideal plane* (that becomes the voxel-fit plane), with the difference between the actual position of the points, p_i , and the position of the projection of these points onto the ideal plane, being signified by the residual, r_i :

$$\alpha(x_i - c_x) + \beta(y_i - c_y) + \gamma(z_i - c_z) = r_i \quad \forall p_i, \quad (4.9)$$

which means that each point, p_i , can be expressed as the sum of the products of a scaling factors, α, β, γ and the deltas, δ_{xyz} :

$$\delta_{x_i} \alpha + \delta_{y_i} \beta + \delta_{z_i} \gamma = p_i. \quad (4.10)$$

Therefore:

$$\begin{pmatrix} \delta_{x_i} & \delta_{y_i} & \delta_{z_i} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = r_i. \quad (4.11)$$

To find the voxel plane, the scaling factors, α, β, γ must be found to minimise the total squared error, $\sum_i^n r_i^2$. Transferring the delta components to matrix form:

$$\begin{bmatrix} \delta_{x_i} & \delta_{y_i} & \delta_{z_i} \end{bmatrix} = \Delta_i \quad \forall \delta_{xyz}, \quad (4.12)$$

leading to the $n \times 3$ matrix, R :

$$\Delta \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = R. \quad (4.13)$$

Finally, this matrix can be calculated such that:

$$\sum_i^n r_i^2 = R^T R = (\alpha \ \beta \ \gamma) \Delta^T \Delta \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = (\alpha \ \beta \ \gamma) \mu \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}, \quad (4.14)$$

where μ is the product of $\Delta^T \Delta$. The eigenvectors can then be extracted from μ , such that ϵ is a resultant 3×3 matrix and the minimum column vector signified by, η , is the normal of the voxel plane:

$$\vec{n} = \begin{bmatrix} \epsilon_{1\eta} \\ \epsilon_{2\eta} \\ \epsilon_{3\eta} \end{bmatrix}. \quad (4.15)$$

The extracted eigenvectors have the property of mutual orthogonality, leading to the remaining column vectors together forming the orthonormal basis of the voxel-fit plane, the remaining orthogonal directions, \vec{u} and \vec{v} , signified by u and v , respectively:

$$\vec{u} = \begin{bmatrix} \epsilon_{1v} \\ \epsilon_{2v} \\ \epsilon_{3v} \end{bmatrix}, \quad (4.16)$$

$$\vec{v} = \begin{bmatrix} \epsilon_{1v} \\ \epsilon_{2v} \\ \epsilon_{3v} \end{bmatrix}. \quad (4.17)$$

4.9 Stage 2: Segmentation through k NN Region Growing(Matlab)

What results from the previous stage is a series of planes, that approximate the surface of point cloud. It is not feasible to take meaningful orientations at this stage because there are too many planes, with their own individual measurements. What is required is the aggregation of all co-planar planes, belonging to the same surface. This replicated throughout the point cloud, will produce a segmented structure that orientations can be suitably derived from [24].

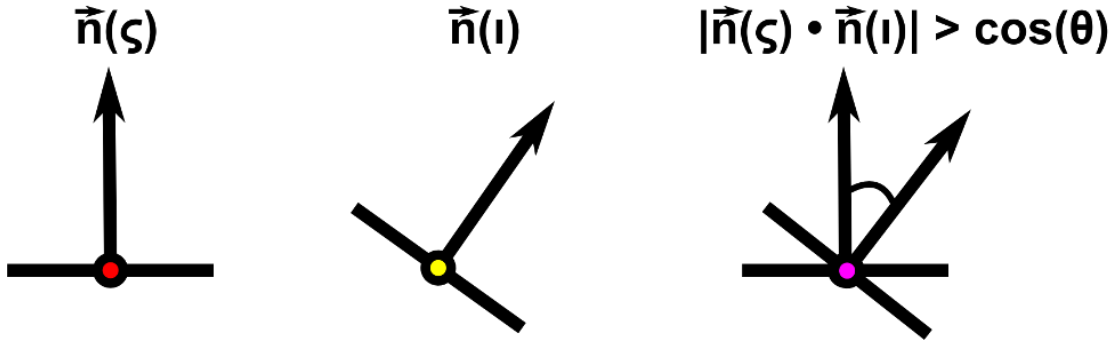


Figure 4.8: Normal angle comparison is achieved through an absolute dot product, that is then compared to the angle $\cos(\theta)$: if it is greater than $\cos(\theta)$, then ι is not part of the region.

Rabbani [48] proposes segmenting a point cloud using a k Nearest Neighbour Region Growing algorithm, that incorporates a threshold metric for the accurate differentiation of edges, and a residual threshold metric for rounded or cylindrical surfaces.

For this work, there are different requirements, to the previous work specified. Rounded formations are unlikely, while separate surfaces with the same normal orientation are likely. The residual threshold should find this case, rather than being optimised for rounded surfaces. This method was implemented as a subroutine using a threshold θ , representing the angle at which deviation from the previous plane is considered a new surface, and surface offset residual threshold, ψ , representing the offset tolerance between planes with similar values for θ .

4.9.1 Large Scale Structure through Region Detection

The purpose of this stage is to detect large scale surfaces and group all like points together in areas of co-planarity. These can then be merged into singular *regions*: large scale planes, from which a representative orientation can be extracted.

Applying the Surface Threshold

Using a similar methodology to Rabbani [48], the author considers there to be two specific cases for differentiation of adjacent voxel planes into different regions. The first is a smoothness constraint angular threshold, θ , where the normal of the reference plane $\vec{n}(\zeta)$ and normal of the candidate plane $\vec{n}(\iota)$ are compared to ensure they are of similar value. If θ is expressed in

radians, this can be calculated through the dot product [48]:

$$|\vec{n}(\zeta) \cdot \vec{n}(\iota)| > \cos(\theta). \quad (4.18)$$

There is a $\frac{\pi}{2}$ value ambiguity regarding the normal direction, so absolute value needs to be taken for the calculation [48].

Applying the Planar Offset Threshold

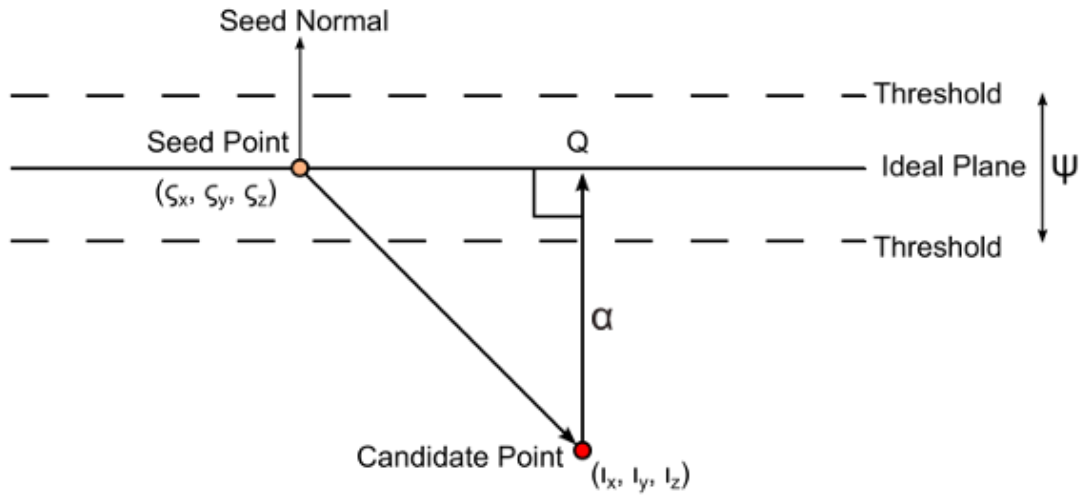


Figure 4.9: Visual description of the ψ planar offset mechanic. Extracting the magnitude at which the ideal plane is intersected, determines co-planarity.

The second case for differentiation is the application of the co-planarity residual offset ψ . This is the case where a candidate plane and the seed plane are not co-joining but the normal vector values for both points are the same.

In this case, outside a specified threshold, planes categorised as parallel, different surfaces. This can be expressed as a Pythagorean problem, where the reference/seed point $S_{(x,y,z)}$ is on the plane and $\iota_{(x,y,z)}$ is some candidate point with a normal vector, \vec{n} , which at some magnitude, α , intersects with the plane, at some point, Q :

$$\begin{pmatrix} Q_x \\ Q_y \\ Q_z \end{pmatrix} = \begin{pmatrix} \iota_x \\ \iota_y \\ \iota_z \end{pmatrix} + \alpha \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}. \quad (4.19)$$

The relationship can be further characterised by describing in the form of the Plane equation:

$$n_x Q_x + n_y Q_y + n_z Q_z + d = 0. \quad (4.20)$$

The projection point Q can be substituted for $\iota + \alpha$:

$$n_x(\iota_x + \alpha n_x) + n_y(\iota_y + \alpha n_y) + n_z(\iota_z + \alpha n_z) + d = 0. \quad (4.21)$$

ζ can be substituted in, to acquire α separately:

$$n_x(\iota_x - \zeta_x) - n_y(\iota_y - \zeta_y) + n_z(\iota_z - \zeta_z) = -\alpha|n|^2, \quad (4.22)$$

$$\alpha|n|^2 = \vec{n} \cdot (\zeta - \iota), \quad (4.23)$$

$$\alpha = \frac{\vec{n}}{|\vec{n}|^2} \cdot (\zeta - \iota). \quad (4.24)$$

Having α as a scalar now allows a unit-less, non-dimensional value, ψ , to be applied as a threshold, such that if α exceeds it, point ι can be evaluated as non co-planar.

4.9.2 Building the Region Plane

Recall that the objective is to build a representation of all voxel planes in the defined region, therefore a *region plane* is a rectangular area fitted to part of the surface. A simple aggregation, or averaging approximation of the voxel planes of the region to create the *region plane*, leads to the aggregation of all residual errors of the voxel plane calculations. This leads to the formation highly erroneous region planes, that often have no spatial relation to the position of the region in the point cloud, or it's dimensions.

The key quantities needed to create a region plane are the same as for any other plane: the centroid C and normal \vec{n} , can be calculated in the same way as for the voxel plane. The best approximation will derive from using only the voxel plane centroids, h , and, only the centroids of all voxel planes in the region, h_i . The region plane centroid is the mean and belongs to the *ideal plane*, I , of the region:

$$c = \bar{h}_i, \quad (4.25)$$

$$\bar{h}_i = c(I) = c(R). \quad (4.26)$$

Again, the first eigenvector column of the centroid points: $c(I)^\top \times c(I)$, produces the ideal plane normal, $\vec{n}(I)$. This allows the calculation of the ideal plane, which is boundless, while the region plane has a boundary, that is the boundary of the region itself. In each case, the voxel plane is definitively bounded by the size of the volume of the search space, v_{ss} . In the case of the region plane, no such boundary exists, therefore, a perimeter must be derived for each region that conforms to its region orientation and dimensions.

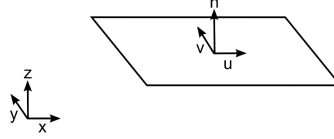


Figure 4.10: The relationship between the world reference and local plane reference

Calculating the Plane Perimeter

It is important to realise in the region plane calculations, that the region points, h_i in the ideal plane are not in reality all in the same plane; there are minor differences in the position of these points with respect to each other and the centroid of the region, especially as the centroid is the mean of all the points in the region. This makes it an *ideal point* and not one of the 'real' points in the set.

This is tackled by defining the ideal plane centroid and projecting all the points of the region onto this plane. Then the outer most boundary can be calculated and a perimeter drawn, by treating the plane in terms of its own coordinate reference frame, defined with the following quantities:

$$o = c(I) = c(R), \quad (4.27)$$

$$loc_x = \vec{u} \text{ direction}, \quad (4.28)$$

$$loc_y = \vec{v} \text{ direction}, \quad (4.29)$$

$$loc_z = \vec{n} \text{ direction}, \quad (4.30)$$

$$(4.31)$$

where the ideal plane centroid is the local origin, \vec{n} is the local z-axis, \vec{u} is the local x-axis and \vec{v} is the local y-axis, and as such, share the quality of mutual orthogonality. Leading on from this, every region point, h_i can be expressed as follows:

$$h_i = o + \beta\vec{u} + \gamma\vec{v} + \rho\vec{n}, \quad (4.32)$$

which expands to

$$\begin{pmatrix} h_{i(x)} \\ h_{i(y)} \\ h_{i(z)} \end{pmatrix} = \begin{pmatrix} o_{(x)} \\ o_{(y)} \\ o_{(z)} \end{pmatrix} + \beta_i \begin{pmatrix} u_{i(x)} \\ u_{i(y)} \\ u_{i(z)} \end{pmatrix} + \gamma_i \begin{pmatrix} v_{i(x)} \\ v_{i(y)} \\ v_{i(z)} \end{pmatrix} + \rho_i \begin{pmatrix} n_{i(x)} \\ n_{i(y)} \\ n_{i(z)} \end{pmatrix}. \quad (4.33)$$

The local z-axis quantity $\rho\vec{n}$ is dispensed with, as the projection of region points h_i on the ideal plane is what is required, and that point exists where $\vec{u} \times \vec{n} = 0$, or $\vec{v} \times \vec{n} = 0$, leading

to:

$$\begin{pmatrix} h_{i(x)} \\ h_{i(y)} \\ h_{i(z)} \end{pmatrix} = \begin{pmatrix} o_{(x)} \\ o_{(y)} \\ o_{(z)} \end{pmatrix} + \beta_i \begin{pmatrix} u_{i(x)} \\ u_{i(y)} \\ u_{i(z)} \end{pmatrix} + \gamma_i \begin{pmatrix} v_{i(x)} \\ v_{i(y)} \\ v_{i(z)} \end{pmatrix}. \quad (4.34)$$

Here, the majority of the quantities are previously known. Only the scalar factors β and γ for each ideal plane point are needed:

$$\begin{pmatrix} \beta \\ \gamma \\ \rho \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{v} \\ \vec{n} \end{pmatrix} = (h_i - o), \quad (4.35)$$

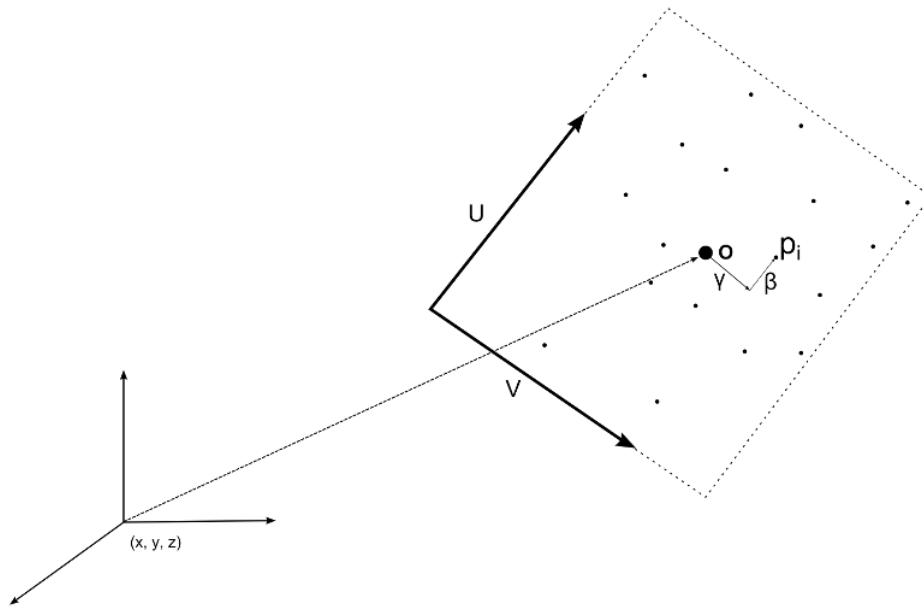


Figure 4.11: The factors β and γ describe the magnitude of direction of the point, p_i , from the origin, o , in the (u, v) coordinate system

$$\begin{pmatrix} \beta \\ \gamma \\ \rho \end{pmatrix} = \begin{pmatrix} \vec{u} \\ \vec{v} \\ \vec{n} \end{pmatrix}^{-1} \cdot (h_i - o). \quad (4.36)$$

Minimum Perimeter Golden Section Search

In bounding the ideal plane, there are some considerations: the number of sides of the region plane is unknown and the orientation of the region plane is also unknown. For the sake of

simplicity, the assumption is made that the region plane always represents the region as a rectangle. This bounds the ideal plane, but the difference between the most efficient area coverage will vary considerably with rotation. What is required, is the perimeter corresponding to the rotation that minimises this area to the greatest degree.

Applying the Rotation Matrix

By starting off with an ideal point in \vec{u} on the ideal plane, a rotation matrix can be used to rotate the ideal plane about the normal axis, \vec{n} , and use the above equations to calculate the β and γ for each rotation.

The new rotation \vec{v} and thus, a new β and γ can be calculated for each point in the set, and the perimeter derived as a scalar. These can be used as scores to be compared between rotations, and the smallest selected as the region plane boundary. As the equation of the plane can be represented as:

$$n_x p_x + n_y p_y + n_z p_z + d = 0, \quad (4.37)$$

an ideal point, t , on the \vec{u} can be found by:

$$x = \frac{-d}{n_x}, \quad \text{where } y = z = 0, \quad (4.38)$$

$$\vec{u}(t) = \left(\frac{-d}{n_x}, 0, 0 \right). \quad (4.39)$$

Thus, the \vec{u} can be derived from the following calculation:

$$\vec{u} = \frac{(\vec{u}(t) - o)}{|\vec{u}(t) - o|}, \quad (4.40)$$

\vec{u} is crossed with the Rotation Matrix to produce a new vector in the u direction, \vec{u}_θ :

$$\vec{u}_\theta = \vec{u} \times M_{rot}(\hat{\omega}, \theta), \quad (4.41)$$

where M_{rot} is Rodrigues' formula utilising the skew symmetric matrix $\hat{\omega}$, and the rotation angle θ :

$$M_{rot}(\vec{n}, \theta) = e^{\hat{\omega}\theta} = I + \hat{\omega} \sin \theta + \hat{\omega}^2 (1 - \cos \theta), \quad (4.42)$$

where I is the 3×3 Identity matrix and $\hat{\omega}$ is the skew symmetric product with \vec{n} :

$$\hat{\omega} = skew(\vec{n}) = \begin{pmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{pmatrix}. \quad (4.43)$$

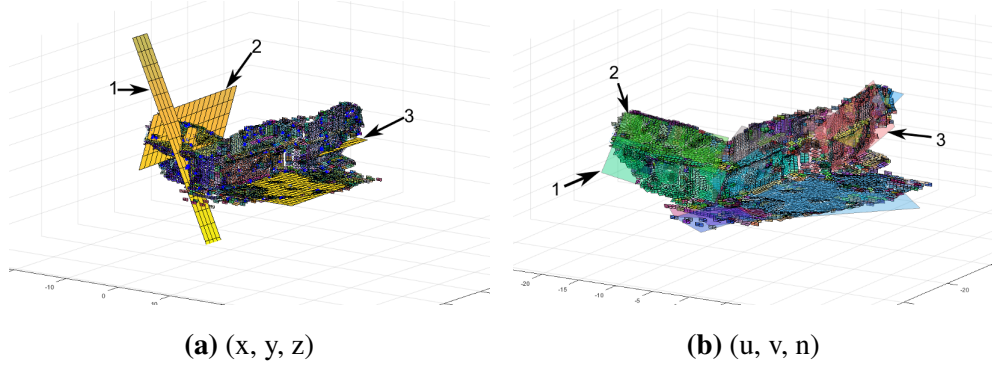


Figure 4.12: The results of using standard World reference for drawing planes(a) and using the φ Golden Section Minimum Perimeter method(b). The same regions are labelled between both. (b) is visibly more accurate.

\vec{v} is the result of the cross product of \vec{u} with the \vec{n} :

$$\vec{v} = \vec{u} \times \vec{n}. \quad (4.44)$$

Given β_i, γ_i , the bounding box of the ideal plane, and therefore, the definite perimeter of the region plane can be calculated using:

$$B = 2(\max(\beta_i) - \min(\beta_i)) + 2(\max(\gamma_i) - \min(\gamma_i)) \quad (4.45)$$

B is a scalar that can be used as a score, computed for every angular change of theta. This process can now be folded into a φ Golden Section Search, and iterated to find the optimum perimeter of the region. Assuming that the region plane is always a rectangular flat surface, that outer boundary is finally characterised by:

$$\vec{U}_{max} = \beta_{max} \vec{u}_{\theta}, \quad (4.46)$$

$$\vec{U}_{min} = \beta_{min} \vec{u}_{\theta}, \quad (4.47)$$

$$\vec{V}_{max} = \gamma_{max} \vec{v}_{\theta}, \quad (4.48)$$

$$\vec{V}_{min} = \gamma_{min} \vec{v}_{\theta}, \quad (4.49)$$

$$(4.50)$$

where $\vec{U}_{max}, \vec{U}_{min}, \vec{V}_{max}, \vec{V}_{min}$, correspond to the coordinates that describe the maximum and minimum extent of ideal plane in each direction. To characterise the boundary, the corners

must be found:

$$I_{\vec{U}_{max}, \vec{V}_{max}} = o + \vec{U}_{max} + \vec{V}_{max}, \quad (4.51)$$

$$I_{\vec{U}_{max}, \vec{V}_{min}} = o + \vec{U}_{min} + \vec{V}_{min}, \quad (4.52)$$

$$I_{\vec{U}_{min}, \vec{V}_{max}} = o + \vec{U}_{min} + \vec{V}_{max}, \quad (4.53)$$

$$I_{\vec{U}_{min}, \vec{V}_{min}} = o + \vec{U}_{min} + \vec{V}_{min}. \quad (4.54)$$

Finally, the region plane, R , can be fully described by the following matrix:

$$R = \begin{pmatrix} I_{\vec{U}_{max}, \vec{V}_{max}} & I_{\vec{U}_{max}, \vec{V}_{min}} \\ I_{\vec{U}_{min}, \vec{V}_{max}} & I_{\vec{U}_{min}, \vec{V}_{min}} \end{pmatrix}. \quad (4.55)$$

4.10 Stage 3: Obtaining Fracture Orientations

Once region planes, with centroid points and normals have been derived, the calculations of orientations in a geological type are well documented and relatively simple to apply.

In this work, the application of the direction cosine method is favoured [24, 19]. The author favoured the application of Feng et al. [19] (fully described in section 3.6) because of the completeness in description of the method. The only deviation being in the derivation of the normal vector. Instead, for this an eigenvector matrix transpose multiplication described above in this section, was used because as this was far more efficient when handling a volumes of variable numbers of points.

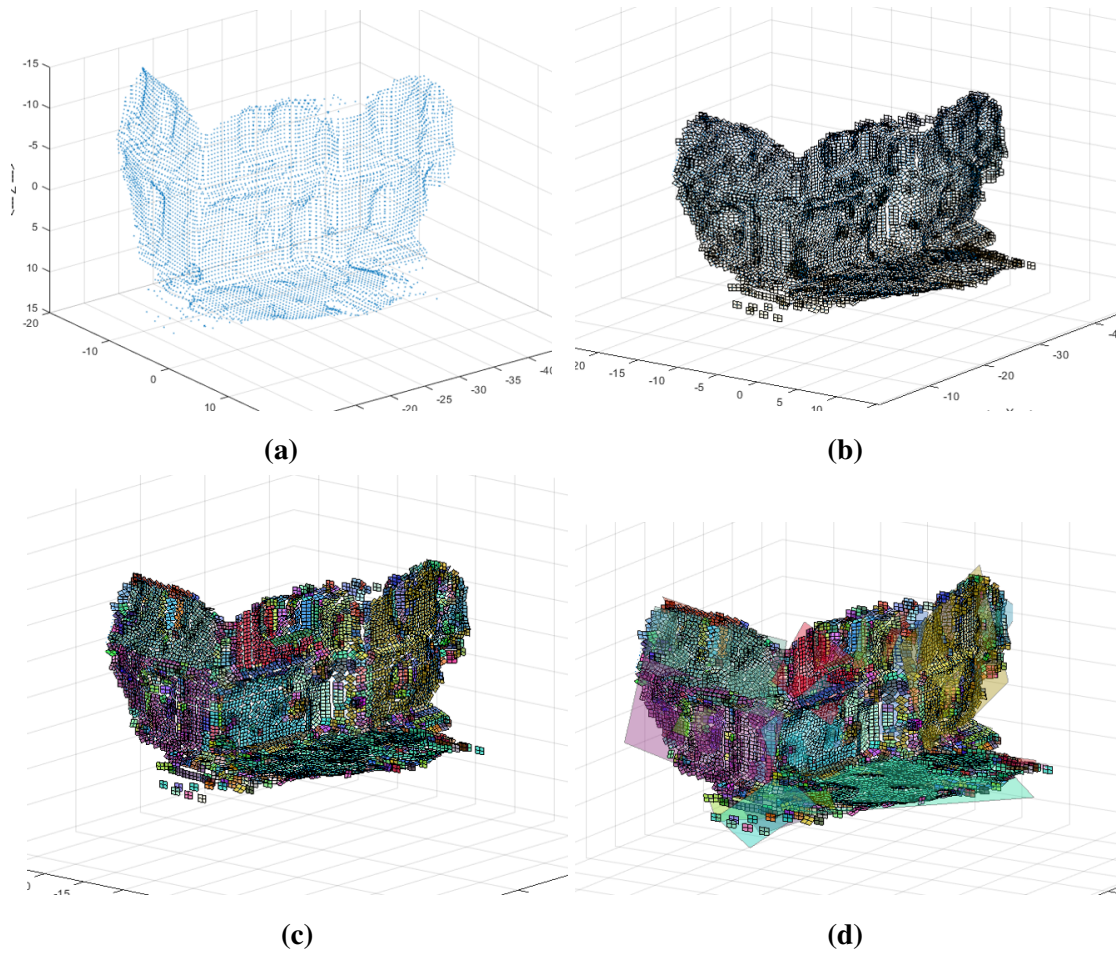


Figure 4.13: The full process of the GeoStructure pipeline is illustrated above, with each of the main stages of point cloud operations shown. (a) shows in input point cloud, after a Mahalanobis filter is applied; (b) shows the structure after the voxel fit process; (c) shows the result of the region growing process, with different region denoted by different colours; (d) illustrates post algorithmic completion, with the regions each aggregated into single planes with composite orientations.

Chapter 5

Results and Analysis

This chapter concerns the evaluation of the Software Pipeline’s performance under test conditions. This section will examine the quality of reconstruction, comparing the original object or scene with the reconstruction. This more qualitative investigation will serve to demonstrate in general terms the result of using the Toolchain in the extraction of large scale structure, using video. The second will examine the accuracy of fracture orientations obtained from the reconstruction as compared to ground-truth measurements.

5.1 General Rationale

A commercial digital camera was used to simulate a camera [on a UAV drone platform] in data collection. This was moved parallel to the building maintaining a distance of 3-4 meters. A specific flight path was chosen, focusing on just changing the height of the camera center (by approximately a 1.5 metres), rather than any more exotic strategies possible [3, 59, 52, 61, 15, 32, 31].

Oblique angle strategies were almost entirely devised for acquisition methods using cameras in a purely photographic configuration. For this simple initial case, where video is the primary sensing



(a)



(b)

Figure 5.1: The top image shows the subject of the experiment: The Hume Cronyn Memorial Observatory; the lower image shows the data collection instrument: a standard 8.0 Mpixel Canon Ixus camera.

source, this methodology was not pursued as the use of video theoretically makes the number of data points much higher. Figure 2.4 shows the flight path of data acquisition. For this demonstration, it is deemed that this is a reasonable facsimile of a drone acquiring video data from a large scale outcrop-type feature.

5.2 Subject Choice Rationale

An easily accessible structure was used to simulate a geological feature, with dimensions and measurements that can be also easily be verified. The Hume Cronyn Memorial Observatory building was chosen as the subject because the general size would be comparable to the types of structures a geologist would find in the field, and also has several features considered useful for the test set:

- Outer circumference is easily traversable so full 3D aspect is possible with a small sample set,
- A number of distinctive minor features that could indicate success of differentiation of algorithm,
- A number of similar large scale features that can serve to demonstrate accuracy of algorithm.

5.3 The Dataset

The dataset consisted of 1 min 25 secs of video in .avi format, taken with the Canon Ixus 8.0 Mpixel in video mode at 640×480 pixel resolution. This decomposed into 2573 individual frames, at approximately 30 fps, so this formed the sample set of input images to the VSfM package.

Several data points were obtained from the subject [building]. These consisted of distance and angular measurements of the walls and structural features of the subject for use as a ground-

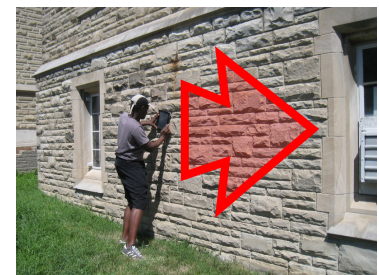


Figure 5.2: Top to bottom: Ground-truth measurements were taken at the centre of the surface; the measurement is recorded; Γ_c direction is always to the right.

truthing comparison. This provides one aspect of evaluation, being the determination of the accuracy of the detail of the reconstruction.

5.4 Result Set Considerations

The first aspect of this result set is to use the Software Toolchain to reproduce a *representative* reconstruction of the subject, and this is primarily evaluated through visual inspection. The second aspect of the result set concerns how *accurate* of the reconstruction of the subject is, this of importance with regard to application of this system in a practical geological context. While not intended to be a replacement for other precise characterisation methods, practical application requires a level of quantifiable accuracy.

The number and configurations of characterised regions the reconstruction, vary mainly dependant on the four factors to the GeoStructure algorithm pipeline. As a brief recap, those variables are:

- ζ : the voxel size factor (dictates the size of the voxel planes for each region),
- θ : the angle threshold (determines if voxel planes are the same surface [for region growing]),
- ψ : the residual offset (determines whether voxel planes are the same surface [for region growing]),
- k : the nearest neighbour value used in the region growing subroutine.

Theoretically each of these should have an optimum number. The second result set aspect consists of a search of the variable space, to determine this optimum for each. As one varies, the others will be maintained at the default values used in the algorithm prototyping and development phase. The development values for each of the variables are as follows:

- $\zeta = 0.01$,
- $\theta = 10$,
- $\psi = 0.1$,
- $k = 15$.

5.5 Ground Truth

A subset of the reproduced regions will be used as a ground truth. As the main interest of the Toolchain is in discerning macro scale structure, mostly the largest, readily reachable surfaces were chosen to obtain fracture orientation measurements from.

The diagram below illustrates which surfaces were chosen and the table illustrates the ground-truthed properties of each. One of the benefits of choosing a subject such as this, is the regularity of the measurements. For instance, it can be logically surmised that surfaces 1-5 should all have a Γ_α of approximately 90° , so this allows an easy intuitive check of both the ground-truth data and the processed fracture orientation output. Surface 6 has an angular geometry, which allows the testing of the accuracy of the Γ_α component. Fortuitously, the long side of the building moves in the North direction, as the long side of the reconstruction moves in the x-axis direction. Because the fracture orientation derivation strategy used considers the x-axis direction as North [19], there is no need for an angular conversion between the ground truth data and the measured data.

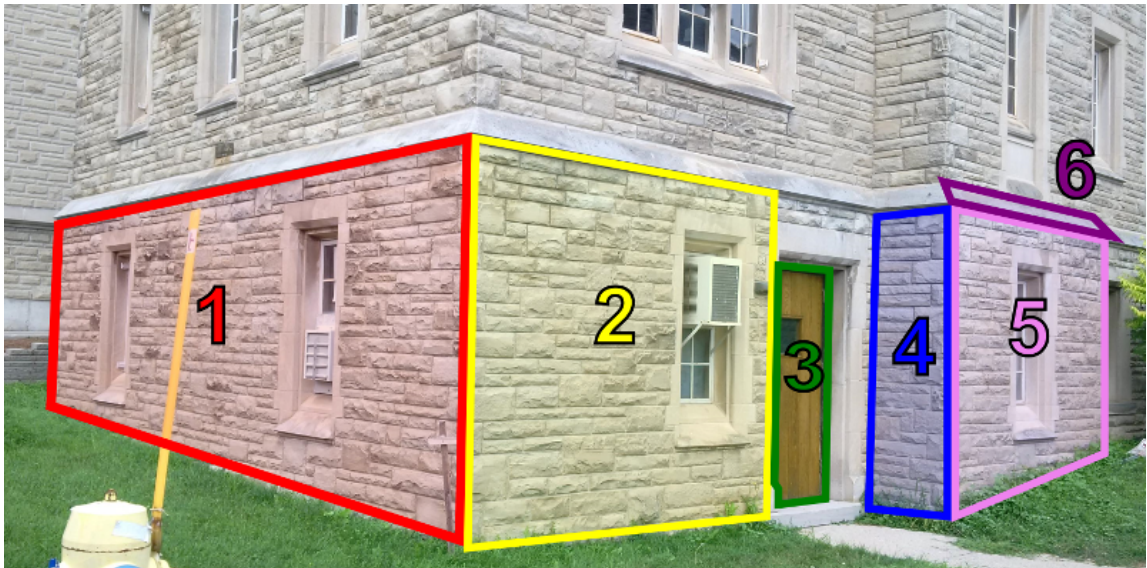


Figure 5.3: The 6 surfaces selected for ground truth measurement are highlighted in this image. The fracture orientations are detailed in the table below.

5.6 Obtaining Measurements

In this case, ground-truth measurements were obtained using the *Strike and Dip* application, developed by *Hunt Mountain Software*, installed on a tablet device, as described by fig. 5.2. From the top to the bottom image: measurements are obtained by laying the device flat on the surface, with the screen facing the user; the interface can then be used to record the measurement; lastly Γ_ζ direction is indicated by the arrow, The Γ_ζ direction is always to the right of the device screen.



Figure 5.4: A close-up image of the inclination of surface 6. The upward normal from the horizontal is the Γ_α recorded. In this case, the angle is 40°

5.7 Fracture Orientation Comparison Quality Score

The ground-truthed and measured orientations can be compared via the standard data normalisation method, where the difference between the angles of the ground truth and measured surfaces can be represented as z , a dimensionless score between 0 and 1:

$$z = \frac{(x_i - x_j) - x_{min}}{x_{max} - x_{min}}, \quad (5.1)$$

where x_i and x_j are the ground-truth and measured quantities, and x_{min} and x_{max} are the maximum and minimum possible values in degrees or radians. This method is still problematic in that the subtraction of larger number from a smaller number will result in a negative quantity. This can be rectified with the following modification:

$$z = \frac{\sqrt{(x_i - x_j)^2} - x_{min}}{x_{max} - x_{min}}. \quad (5.2)$$

This structure now always results in a positive fractional number between 0 and 1.

5.8 Measuring Pipeline Factor Efficacy

The efficacy of variation of each of the pipeline factors can be analysed through two methods. Firstly, through the use of a timing system: how long to overall completion of the pipeline,

and what fraction of that is down to region growing, which is the single most complex part. Secondly, the effect of factor variation is analysed through the application of the Comparison Quality Score to each of the six selected region planes. This method works such that a measurement identical to the ground truth gives the value of zero, with larger discrepancies between the measurements resulting in values closer to 1. In order to measure fidelity consistently, surfaces will only be considered if they remain both whole and not over differentiated, so, should a surface split into two, it will automatically be given a score of 1. Similarly, if a surface merges with another to make a larger surface, it is neither of the originals but in fact a new surface and is given a score of 1.

Simulating Fracture Orientations with Rotation Matrices

The solution to the Discontinuity Problem, (as discussed in Section 3.6.3) is to measure the difference between of ground-truth and reconstructed region plane normals. In this case, obtaining ground truth normal measurements requires geometrically reproducing the fracture orientation using the rotation matrix. Specifically, a complete representation of any fracture orientation can be simulated using the rotation matrices of the z-axis and the y-axis, assuming the conversion from cartographic coordinates to geometrical coordinates.

Working in the standard cartographic coordinate system, the fracture orientations Γ_ζ , Γ_δ and Γ_α can be conceptualised in 2D as a plane, where the initial position of the plane has the positive strike position pointing West at 270° , the negative strike position points East at 90° and the normal of that surface is the Γ_δ at 0° . A rotation about the z-axis is used to get to the desired Γ_δ value, multiplied by a rotation about the y-axis. The final orientation of the normal, Γ_δ , is extracted as the first column from the resultant matrix.

After translation from map coordinates to geometric coordinates, this method provides an “archetypal” ground-truth measurement, that the normal of the reconstructed surfaces can be measured against.

Measuring Differences Between Vectors

The relationship between any two vectors can be characterised by the squared difference between them. Assuming the vector, \vec{v} , and the vector, \vec{w} , are both normalised to magnitude 1, they can be placed on top of each other and measured to produce a numeric, x_d such that the

maximum possible distance extracted is equal to 2. This comes about when the two normals are diametrically opposed, whereas, a distance of 0 would equate to an exact match.

$$x_d = \sqrt{(v_{(a)} - w_{(a)})^2 + (v_{(b)} - w_{(b)})^2 + (v_{(c)} - w_{(c)})^2}, \quad 0 \geq x_d \leq 2. \quad (5.3)$$

Once the distance is obtained, the Quality Score, z , can be obtained by using the vector distance in place of $x_i - x_j$:

$$R(z) = \frac{x_d - x_{min}}{x_{max} - x_{min}}. \quad (5.4)$$

As before, a dimensionless numeric value between 0 and 1 is produced, where x_{max} is the maximum distance and x_{min} is 0.

This results in a Quality score for a single region surface. The same methodology can be generalised to obtain the overall score for the six regions of that experimental run:

$$z(R) = \frac{\sqrt{\sum_1^n (x_i + \dots + x_n)^2} - x_{min}}{x_{max} - x_{min}}, \quad (5.5)$$

this time, where x_{min} and x_{max} are 0 and n , respectively. This final quantity gives a normalised bounded score for the *run*, which is continuous and directly comparable in all cases. Each *experiment* consists of ten runs, where one of the pipeline factors are varied by a uniform amount. This leads to four separate experiments, with the aim of yielding optimum values for each.

Table 5.1: Table of Ground-truth Values

Surface	Strike(°)	Dip Angle(°)	Dip Direction(°)	\vec{n}
1	87	89(S)	177	-0.9985, -0.0523, 0.0175
2	04	89(E)	94	-0.0697, -0.9974, 0.0175
3	339	87(E)	69	0.3579, -0.9323, 0.0523
4	102	86(S)	192	-0.9970, 0.0348, 0.0698
5	350	89(E)	80	0.1736, -0.9847, 0.0175
6	359	40(E)	89	0.0112, -0.6427, 0.7660

Finally the algorithm for finding the values for each run are simply by dividing the end number by the number of runs and using this as the increment from the starting number. For each pipeline factor, the start, end and (increment) values are listed in below:

- $\zeta = 0.01, 0.07, (0.006)$,

- $\theta = 0, 30, (3)$,
- $\psi = 0.01, 0.6, (0.06)$,
- $k = 1, 20, (2)$.

5.9 Experimental Results

5.9.1 Experiment 1: ζ

Time Series

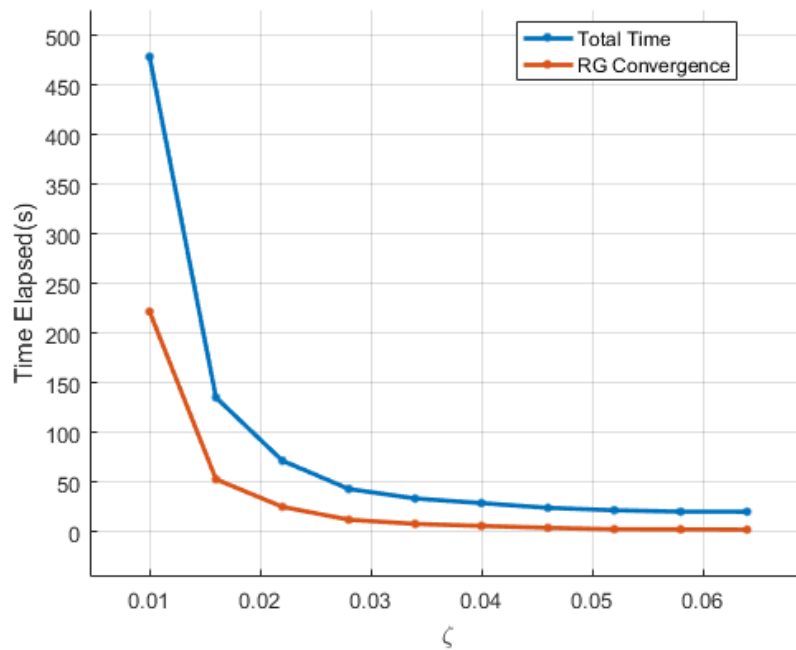


Figure 5.5: ζ v time

From the plotted outputs, it is clear that time to completion and Quality of reconstruction are directly related. The relationship between the ζ quantity and the elapsed time is of logarithmic appearance and the form is almost completely identical, with the distance between the completion of the region growing element and completion of the algorithm being very small.

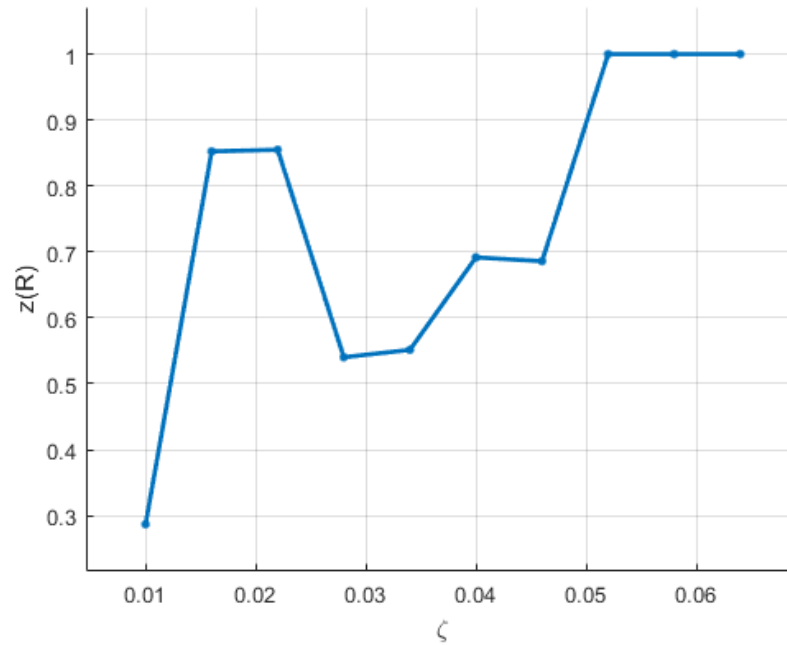


Figure 5.6: ζ v $z(R)$

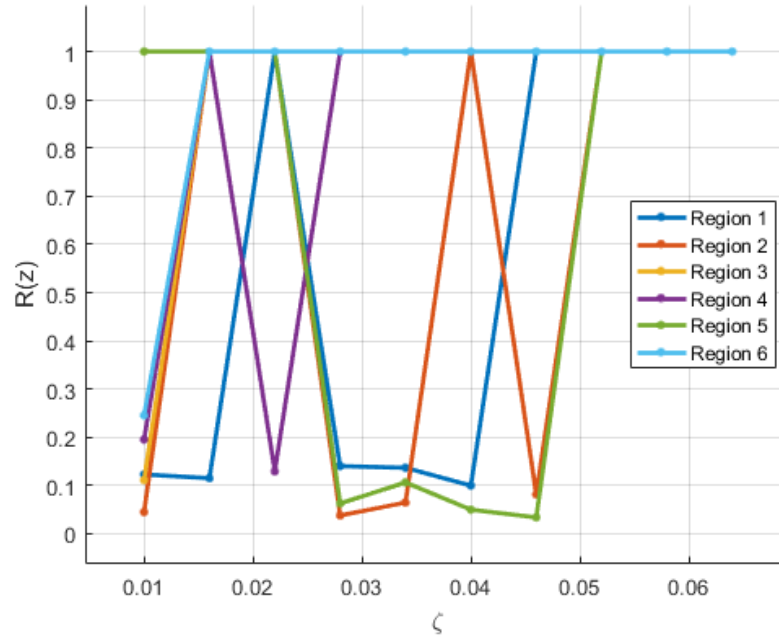
Fracture Orientation Quality Score Analysis: Runs

$z(R)$ show the minimum at the first value in the set, with a general trend towards a value of $z = 1$ with each successive iteration. This trend of z against ζ first increases dramatically to $z = 0.85$ at $\zeta = 0.02$, before falling sharply to $z = 0.5$ for $\zeta = 0.03$. The trend then follows a gradual increase, with a plateau at $z = 1$ starting at approximately $\zeta = 0.05$ and to continuing through the remaining iterations.

Fracture Orientation Quality Score Analysis: Regions

Looking at the region breakdown in terms of Quality, it can be seen that all start within a range from $z = 0$ to $z = 0.4$. There is reasonably quick progression to $z = 1$, with all cases being this by $\zeta = 0.06$, Region 6 and 3 being the first, reaching saturation at $\zeta = 0.02$, while Region 5 being the last at $\zeta = 0.06$.

Looking at the region breakdown in terms of Quality, it can be seen that 5 regions start within a range from $z = 0$ to $z = 0.3$. There is a dramatic oscillation for most regions between $z = 1$ and approximately $z = 0.1$ for $\zeta = 0.01$ to 0.03 , before a steadying at the lower score for $\zeta = 0.03$ to 0.05 . All cases are $z = 1$ by approximately $\zeta = 0.05$.

Figure 5.7: ζ v $R(z)$

Combined Interpretation

Examining the output from the algorithm, this result makes sense, in that ζ is a quantity expressing the proportional size of the search voxel. As the ζ progresses to higher values, the level of detailed evaluation possible will decrease dramatically at larger values for ζ . At $\zeta = 0.1$, the space will be evaluated by only 10 voxels, so this rapid response is predictable. This explains fast convergence to $\zeta = 1$ for Regions 3 and 6: these represent the building door and angular ledge feature and these are both small scale details of the structure, therefore requiring higher levels of differentiation. This also explains the shape of the time curve and the minimal interval between Region Growing completion and total algorithmic completion. The sharp increase in the overall quality between the first and second run indicate some anomalous effect that could be the result of some interplay between the voxel scaling and the specific orientation of the surfaces. This needs further investigation.

5.9.2 Experiment 2: θ

Time Series

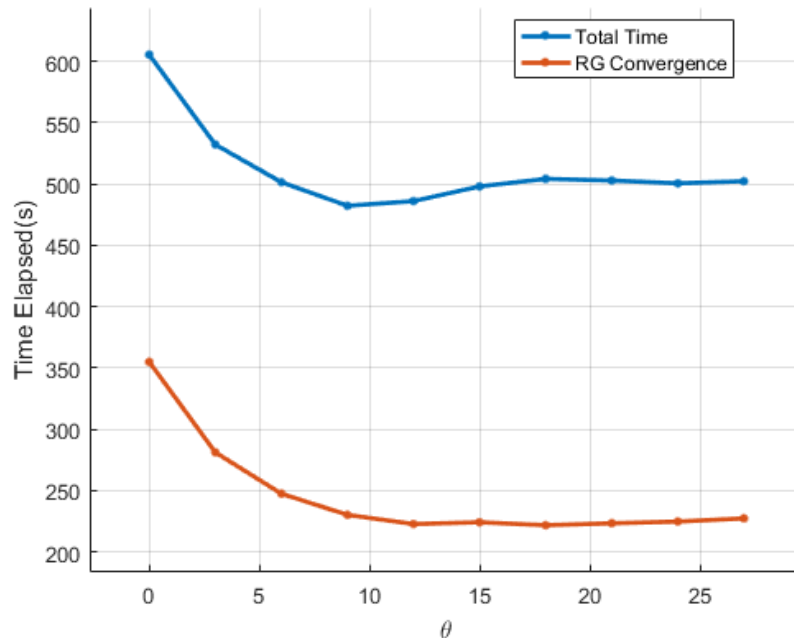
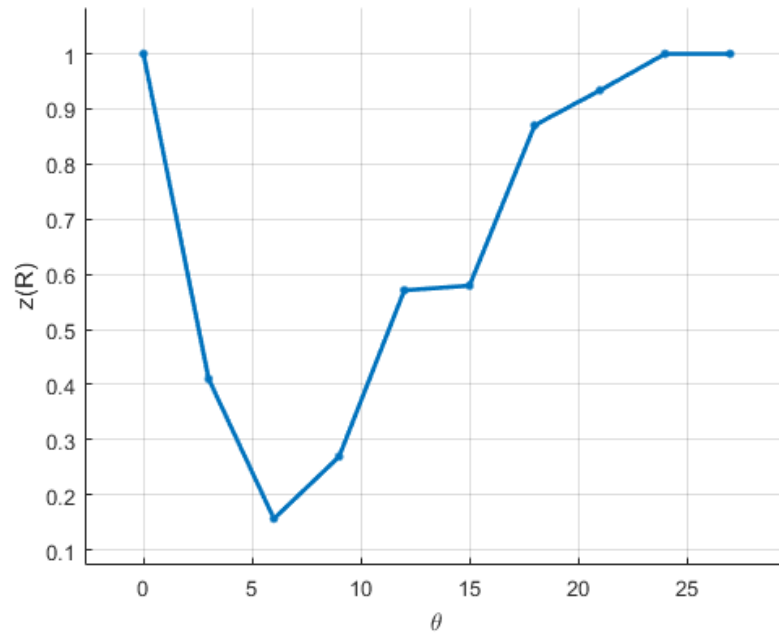


Figure 5.8: θ v time

The relationship between the two curves is almost completely identical, with the distance between the completion of the region growing element and completion of the algorithm being large and uniform across iterations, at generally around 200 secs. The curve shows a gentle decline with a maximum at $\theta = 0$ and a minimum at approximately $\theta = 12$, with following iterations maintaining and identical timing.

Fracture Orientation Quality Score Analysis: Runs

Overall quality shows the maximum score $z = 1$ at the first value in the set, with a trend towards a ground-truth - measurement parity, to a value of approximately $z = 0.15$ before progressively increasing back with each successive iteration achieving to $z = 1$ at approximately $\theta = 24$, and remaining there.

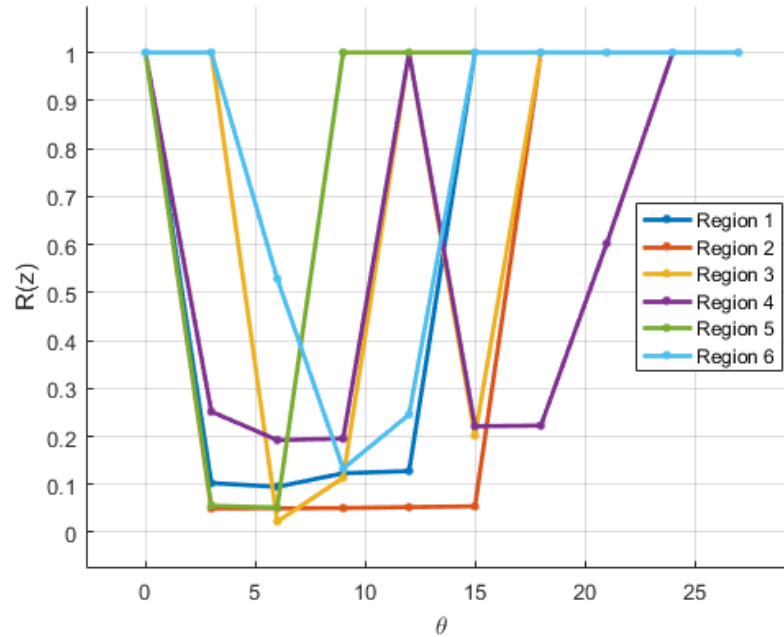
Figure 5.9: $\theta \vee z(R)$

Fracture Orientation Quality Score Analysis: Regions

Looking at the region breakdown in terms of Quality, it can be seen that all start at $z = 1$. There is immediate progression to $z = 0.2$ and lower, with all cases being this returning to $z = 1$ by $\theta = 24$. Regions 4, 3, 5 and 1 drop to $z = 0.1$ and below between $\theta = 4$ to 8, while Region 3 reaches $z = 0$ at $\theta = 6$.

Combined Interpretation

Examining the output from the algorithm, a clear optimum range is discernible between $\theta = 3$ and $\theta = 8$, where the overall quality is between $z = 0.1$ and $z = 0.4$. After this it rises steeply to $z = 0.8$ and then on to $z = 1$. This can be interpreted as the ultra differentiation during region growing for $\theta = 0$, where the slightest difference in normal creates a new region, to a much reduced differentiation for higher values as too many distinct surfaces become recognised as similar regions.

Figure 5.10: θ v $R(z)$

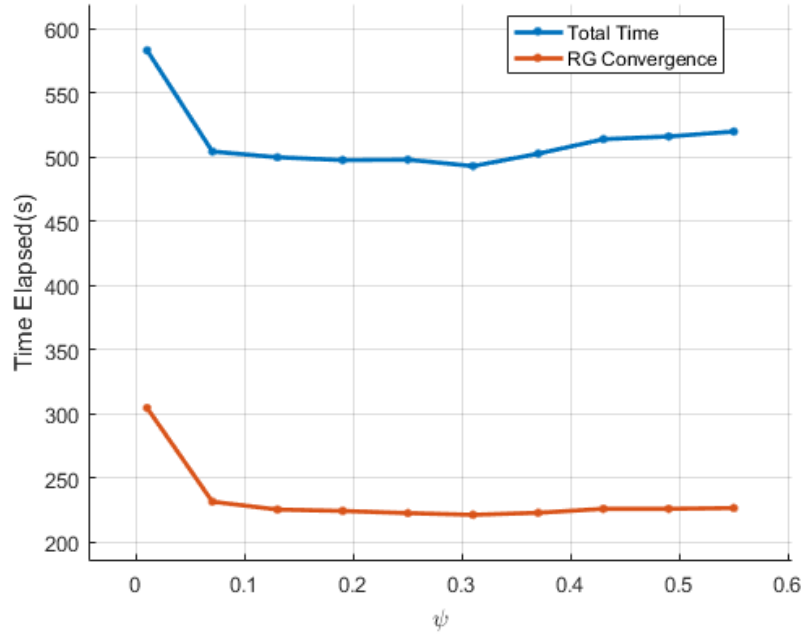
5.9.3 Experiment 3: ψ

Time Series

The relationship between the two curves is almost completely identical, with the distance between the completion of the region growing subroutine and completion of the algorithm being large and uniform across iterations, at approximately 200 seconds. The curve profile maximum at $\psi = 0.01$ with an immediate minimum, followed directly by a general plateau through the following iterations.

Fracture Orientation Quality Score Analysis: Runs

Overall quality shows the minimum score of approximately $z = 0.3$ at the second value of $\psi = 0.07$, with a sharp increase to z over the next few iterations, plateauing out to approximately $z = 0.1$ from $\psi = 0.5$ onwards.

Figure 5.11: ψ v time

Fracture Orientation Quality Score Analysis: Regions

Looking at the region breakdown in terms of Quality, it can be seen that there is a fairly even spread of scores from $z = 1$ to almost $z = 0$. Between $\psi = 0.01$ and 0.3 , quality is generally maintained below the half way point, after which the scores generally increase to $z = 1$ and stay there.

Combined Interpretation

Examining the output from the algorithm, a clear optimum range is discernible between $\psi = 0.01$ and $\psi = 0.2$ where the overall quality is between $z = 0.05$ and $z = 0.5$. After this it rises rapidly to $z = 0.6$ and $z = 0.8$. $\psi = 0.1$ can be interpreted as the optimum distance between planes that can be considered part of the same surface region during region growing. By the time $\psi = 0.4$, this is far enough from any seed plane, that all candidate planes start to become more difficult to distinguish.

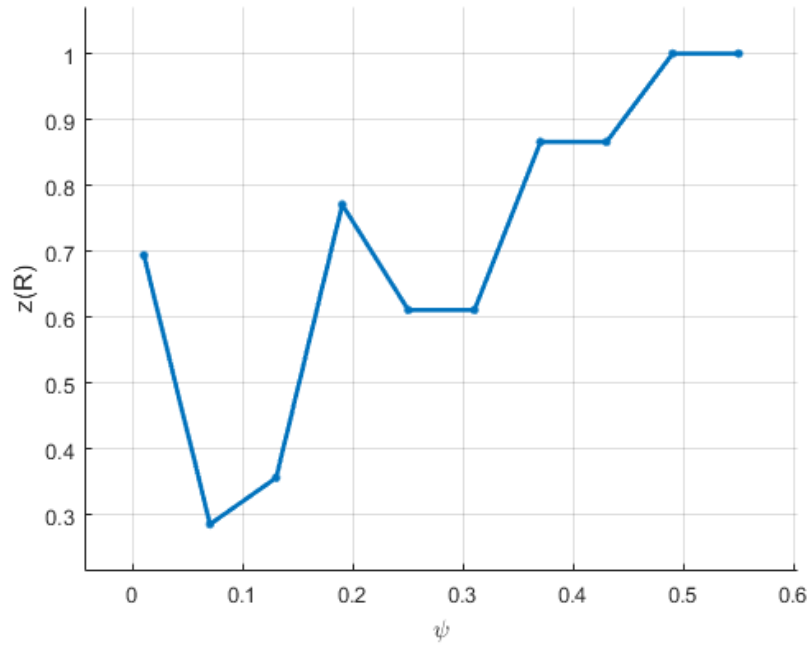


Figure 5.12: ψ v $z(R)$

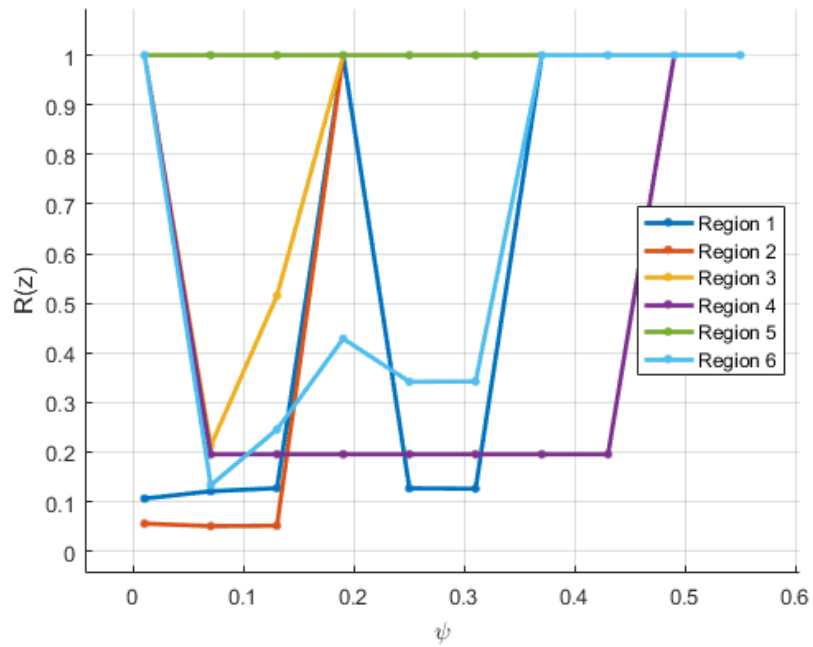


Figure 5.13: ψ v $R(z)$

5.9.4 Experiment 4: k

Time Series

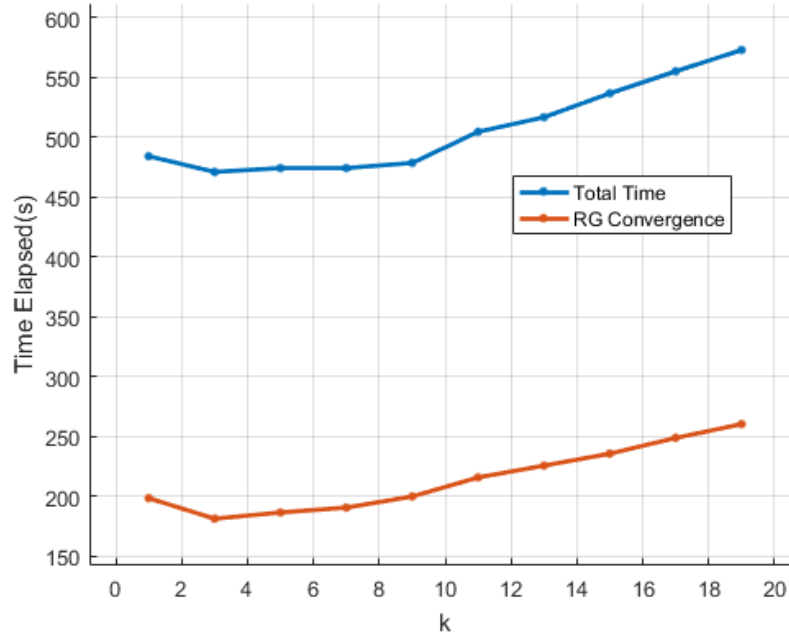


Figure 5.14: k v time

The relationship between the two curves is almost completely identical, with the distance between the completion of the region growing subroutine and completion of the algorithm being large and uniform across iterations, at approximately 200 seconds. The curve profile maximum at $k = 19$ and a minimum at $k = 3$, with a gentle incline from $k = 4$ onwards.

Fracture Orientation Quality Score Analysis: Runs

Overall quality shows the minimum score of just above $z = 0.1$ at the second value of $k = 3$, with two plateaus connected by a steep incline. The first plateau between $k = 3$ and $k = 11$ and the second between $k = 13$ and $k = 19$.

Fracture Orientation Quality Score Analysis: Regions

Looking at the region breakdown in terms of Quality, it can be seen that scores are evenly distributed through out the range. A consistent concentration is visible between $z = 0.2$ and

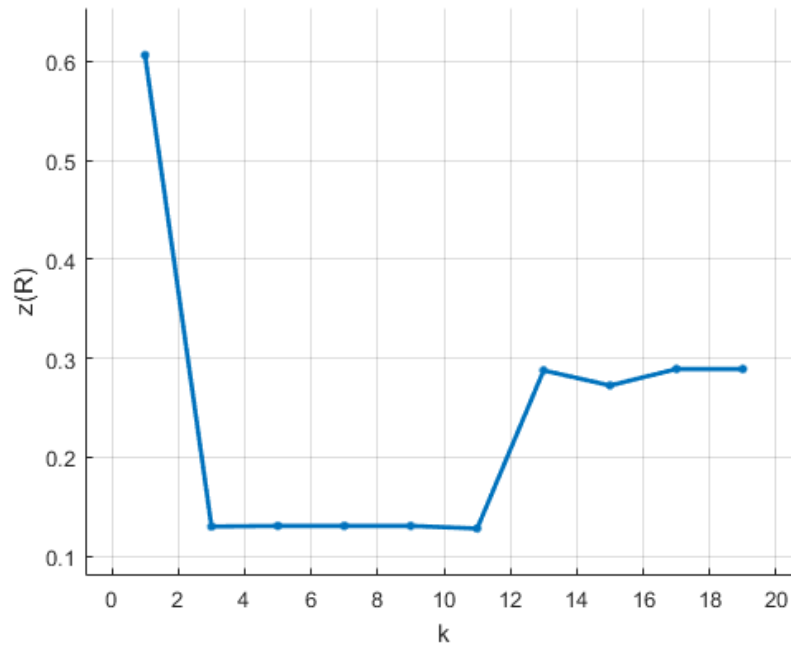


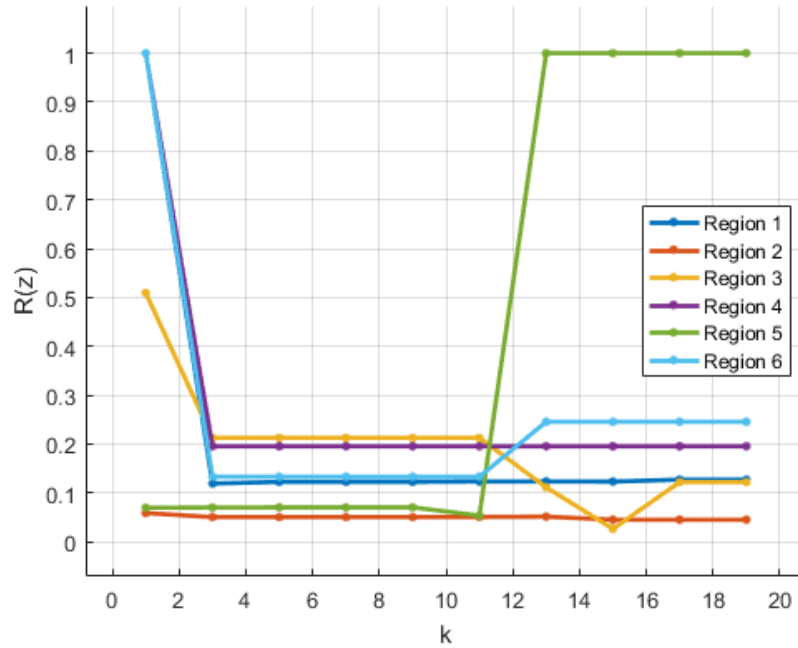
Figure 5.15: $k \vee z(R)$

$z = 0.4$ and this form a structure of note.

Looking at the region breakdown in terms of Quality, it can be seen that scores generally remain very low, between $z = 0$ and $z = 0.2$ for nearly all iterations. Only Region 5 breaks this trend, dramatically rising from below $z = 0.1$ to $z = 1$ in one step at $k = 11$ and $k = 13$.

Combined Interpretation

Examining the output from the algorithm, a clear optimum point is between $k = 3$ and $k = 11$ where the overall quality is just above $z = 0.1$. After this it rises to $z = 0.3$ dramatically and plateaus there. This can be interpreted as a smaller k allowing for more differentiation between planes and segmentation, helping to ensure that during region growing, candidate points do not “jump” over smaller delineating features and grow regions in areas that they should not be. k is a quantity that works very synergistically with the the other factors, so generally, with the exception of extreme values, a vast difference in different values k should not be expected.

Figure 5.16: $k \vee R(z)$

5.9.5 Overall Evaluation

It is clear to see that in all cases more optimal results, that is z values between 0 and 0.4, occur in almost universally at lower values for each variable space. It is also noticeable that all variables with the exception of ζ , start at a higher z before immediately dropping to a much lower z , then rising again (usually at a more gradual interval). This indicates the potential optimum values for z in these cases.

Generally the relationship between the Region Growing completion and algorithm completion is uniform and self-similar, again, with the exception of ζ , where a much greater time elapses for low values, than higher values and the difference between curves for Region Growing and algorithm completion are much less, both in size and uniformity.

Taking these individual results together, it can be concluded that low values for each of the variables are optimal, with the resultant increased time constant of the algorithm being significant relative to higher values but in absolute terms being approximately 500 sec. An interesting progression in the experiment would be to see if optimal values for each variable result in much different timings or z scores.

Chapter 6

Conclusion and Future Work

This chapter is comprised of the following sections: Section 1 will provide a complete summary of the thesis composition from motivations to the results. Section 2 will outline the contributions made by the author, and Section 3 will discuss areas for improvement and/or future work to be undertaken in this area.

6.1 Motivation

The motivation for this work came from the observation of potential areas of improvement and modernisation in several areas:

There are difficulties and challenges associated with the acquisition, processing and interpretation of fracture orientation measurements. Factors such as coverage of measurements, speed of acquisition, processing and interpretation converge to limit practical application of this technique in understanding and characterisation of large-scale structure.

Automation provides both benefits and drawbacks regarding the application of Lidar and laser ranging technologies in the field. There are definite improvements in the ability to characterise, and derive measurements at all scales, that have a beneficial effect on the characterisation of large features, such as craters, and, the reliably high fidelity in reconstruction provides the potential of doing this *automatically*, with very high accuracy. However, there are also challenges such as: cost, the safety implications of using high powered laser equipment, and, equipment practicality and robustness in the potentially varied and extreme conditions in the field.

Regardless of the shortcomings, there are definite benefits to applying a Lidar-type data acquisition strategy to the characterisation of fracture geometries, especially with constant improvement in computing power making the processing of high density data more practically applicable, than in previous years. In spite of this, the benefits to applying a Lidar-type strategy to the problem of acquiring and characterising fracture orientations are clear, especially with improvements in computer power and data processing strategies.

This leads to the question of whether there was a methodology that can produce a similar result without the shortcomings, and a potential solution emerges from the fields of geological photogrammetry, multi-view geometry and SfM, where algorithmic improvements and innovations now allow for proven sub-decimetre precision [27].

While this is much lower resolution than even standard low quality Lidar sensing, it is still more than adequate for large-scale characterisation. Hardware requirements to acquire this precision are orders of magnitude lower than even entry level Lidar-type equivalents, while simultaneously engineered to be far more compact, flexible and resistant to environmental conditions.

The adoption of UAV and drone technology into the area of geological survey activities, suggests potential for an convergence, to produce a single coherent strategy for the acquisition of fracture orientations. Utilising the same methodology for Lidar techniques and incorporating fully automated interpretation of fracture orientation measurements. The goals were therefore to:

- Create a system for direct orientation extraction from the point cloud structure,
- Make calculation and interpretation of the measurements fully automatic.

Other image based extraction methods (even SfM based methods) used techniques to either infer three-dimensional structure from image sources, or employ user interaction to characterise fracture orientations for processing [59, 56, 28, 3]. This was a clear difference in application with Lidar, for instance, where sheer data density meant that measurements can be directly measured from the resulting point cloud. This methodology was replicated through use of video, where the splitting into individual frames at approximately 25 fps to systematically increase the number of unique angles, and images (therefore samples) of the scene, increasing the data density and [theoretically] the fidelity of the reconstruction.

6.1.1 Mitigating the Software Generated Point Cloud Noise Problem

The problem of noise generation through the inevitable introduction of systematic errors using the software generated methods was mitigated, first through the extraction of a low noise, high fidelity surface via a PSR tessellation methodology; then through the application of a Mahalanobis distance metric, that was more successful in the preservation of legitimate structure, while removing extraneous points, than a standard least squares method for outlier removal.

6.1.2 Development of the GeoStructure Algorithm Pipeline

Finally, the orientation extraction was achieved by the development of an algorithmic chain that extracts small scale structure and aggregated that into large scale structure. This required several stages, involving the integration of multiple techniques including a voxel evaluation and plane fit method to characterise the modified point cloud in the first instance, implementation of a k NN Region Growing to aggregate voxel planes into large scale *regions* and a Minimum Perimeter Golden Section plane bounding strategy, to spatially represent and characterise region visualisation with accuracy.

This enabled the implementation of a field geology proven automatic trigonometric technique, utilising direction cosines to measure final orientations of large scale region, with the entire process from input of tessellated point cloud, to extraction of fracture orientations requiring no user interaction.

6.2 Areas for Improvement and Future Work

The variable testing implemented for the Results section could be further explored, with ranges of variables being lengthened and samples within those ranges being increased. This could definitely define optimum values for the pipeline factors. Similarly, the further examination of the PSR octree options could provide an initial point cloud to the GeoStructure algorithm of a higher fidelity, therefore allowing for higher potential fidelity in final output.

Although the GeoStructure algorithm operate completely without user intervention, the Software Toolchain is not fully automatic. User interaction is required in the preprocessing stage, from frame separation to PSR output. This part of the processing chain used the VSfM and Meshlab applications. Both these applications are capable of being run using command line

instructions, or configuration scripts to automate simple functions. In terms of the Toolchain, the requirements for those applications are not complex, so this is an implementable change. The result would be a fully automatic process, requiring absolutely no user input from commencement to completion.

Processing times were manageable for the test dataset, but for surveys involving a much larger dataset, processing times would increase significantly, as the time cost increases exponentially for the number of views in the reconstruction [62, 63]. Further investigation of a range of dataset sample sizes for time complexity is an avenue of exploration.

The use of High Performance Computing architectures, such as Field Programmable Gate Arrays (FPGAs) and Graphical Processing Units (GPUs) hold potential in the exploration of this problem. These architectures are proven effective in various areas when problems can be specifically defined and tailored to the hardware. Some work has already been completed in the application of core elements of the SfM algorithm to High Performance Computing (HPC) platforms [47], further work in this direction could lead to interesting results.

The visual representation of the region plane segmentation requires further development. Accurate and easily understandable visual representation will be necessary, if to be used in the field by operators without a technical background. Work can still be done to more accurately represent the region shape and orientation.

Although covered in the literature, further examination of flight plan strategies is a useful direction. Explicitly collecting data at a much higher sample rate, with more unique angles could require the development of specific strategies to maximise efficiency and fidelity in reconstruction. Is there a maximum sample rate at which no further quality benefits are possible, for instance?

Practical testing of natural geological features and then, field testing with drone technology in a geological setting should remain the ultimate goal.

Bibliography

- [1] S.M. Adams and C.J. Friedland. A Survey of Unmanned Aerial Vehicle (UAV) Usage for Imagery Collection in Disaster Research and Management. *9th International Workshop on Remote Sensing for Disaster Response*, 2011.
- [2] J.A. Bellian. Digital Outcrop Models: Applications of Terrestrial Scanning Lidar Technology in Stratigraphic Modeling. *Journal of Sedimentary Research*, 75(2):166–176, 2005.
- [3] Sean P. Bemis, Steven Micklethwaite, Darren Turner, Mike R. James, Sinan Akciz, Sam T Thiele, and Hasnain Ali. Ground-based and UAV-Based photogrammetry: A multi-scale, high-resolution mapping tool for structural geology and paleoseismology. *Journal of Structural Geology*, 69:163–178, 2014.
- [4] Dimitris Bouris, Antonis Nikitakis, and Ioannis Papaefstathiou. Fast and efficient FPGA-based feature detection employing the SURF algorithm. *Proceedings - IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM 2010*, pages 3–10, 2010.
- [5] M.A. Brophy. Surface Reconstruction from Noisy and Sparse Data. *UWO Electronic Thesis and Dissertation Repository*, Paper 3375, 2015.
- [6] M. Brown and D. G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, pages 56–63, 2005.
- [7] M. Brown and D.G. Lowe. Invariant Features from Interest Point Groups. *British Machine Vision Conference, BMVC 2002*, pages 656–665, 2002.
- [8] F. Calakli and G. Taubin. SSD: Smooth signed distance surface reconstruction. *Computer Graphics Forum*, 30(7):1993–2002, 2011.

- [9] Ninghua Chen, Nina Ni, Paul Kapp, Jianyu Chen, Ancheng Xiao, and Hongge Li. Structural Analysis of the Hero Range in the Qaidam Basin, Northwestern China, Using Integrated UAV, Terrestrial LiDAR, Landsat 8, and 3-D Seismic Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(9):4581–4591, 2015.
- [10] Javier Civera, Andrew J. Davison, and José María Martínez Montiel. *Structure from Motion using the Extended Kalman Filter*, volume 75 of *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [11] Javier Civera, Oscar G. Grasa, Andrew J. Davison, and J. M. M. Montiel. 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, may 2010.
- [12] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. *ACM Transactions on Graphics*, 23:905, 2004.
- [13] Alexandru Constantiniu, Paul Steinmann, Tom Bobach, Gerald Farin, and Georg Umlauf. The Adaptive Delaunay Tessellation: A neighborhood covering meshing technique. *Computational Mechanics*, 42(5):655–669, 2008.
- [14] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- [15] Elisa Dall’Asta, Klaus Thoeni, Marina Santise, Gianfranco Forlani, Anna Giacomini, and Riccardo Roncella. Network Design and Quality Checks in Automatic Orientation of Close-Range Photogrammetric Blocks. *Sensors*, 15(4):7985–8008, 2015.
- [16] Tirtharaj Dash. Automatic navigation of wall following mobile robot using Adaptive Resonance Theory of Type-1. *Biologically Inspired Cognitive Architectures*, 12:1–8, 2015.
- [17] Chris Engels, H Stewénus, and David Nistér. Bundle adjustment rules. *Photogrammetric computer vision*, pages 266–271, 2006.
- [18] Peter L Falkingham. Acquisition of high resolution three-dimensional models using free, open-source, photogrammetric software. *Palaeontologia Electronica*, 15(1):1–15, 2012.
- [19] Q. Feng, P. Sjögren, O. Stephansson, and L. Jing. Measuring fracture orientation at exposed rock faces by using a non-reflector total station. *Engineering Geology*, 59(1-2):133–146, 2001.

- [20] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [21] E. Fix and J.L. Hodges. Discriminatory Analysis Nonparametric Discrimination. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.
- [22] Jan-Michael Frahm, Marc Pollefeys, Brian Clipp, David Gallup, Rahul Raguram, ChangChang Wu, and Christopher Zach. 3D Reconstruction of Architectural Scenes From Uncalibrated Video Sequences. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 38(5/W1):7, 2009.
- [23] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.
- [24] Giovanni Gigli and Nicola Casagli. Semi-automatic extraction of rock mass structural data from high resolution LIDAR point clouds. *International Journal of Rock Mechanics and Mining Sciences*, 48(2):187–198, 2011.
- [25] M. Goesele, B. Curless, and S.M. Seitz. Multi-View Stereo Revisited. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:2402–2409, 2006.
- [26] Christopher Gomez. Multi-scale Voxel-based Algorithm for UAV-derived Point-clouds of Complex Surfaces. pages 205–209, 2014.
- [27] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd edition, 2003.
- [28] Steve Harwin and Arko Lucieer. Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from Unmanned Aerial Vehicle (UAV) imagery. *Remote Sensing*, 4(6):1573–1599, 2012.
- [29] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *ACM*, 26(2):71–78, 1992.
- [30] J Hu, S You, U Neumann, and K K Park. Building Modeling from LIDAR and Aerial Imagery. *Asprs*, 4:23–28, 2004.

- [31] M. R. James and S. Robson. Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application. *Journal of Geophysical Research: Earth Surface*, 117(3):1–17, 2012.
- [32] Mike R. James and Stuart Robson. Mitigating systematic error in topographic models derived from UAV and ground-based image networks. *Earth Surface Processes and Landforms*, 39(10):1413–1420, 2014.
- [33] Michael Kazhdan and Hugues Hoppe. Poisson surface reconstruction. *Proceedings of the fourth Eurographics symposium on Geometry processing*, 7, 2006.
- [34] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013.
- [35] John Kemeny and Randy Post. Estimating three-dimensional rock discontinuity orientation from digital images of fracture traces. *Computers and Geosciences*, 29(1):65–77, 2003.
- [36] Yi Lin, Miao Jiang, Yunjun Yao, Lifu Zhang, and Jiayuan Lin. Use of UAV oblique imaging for detection of individual trees in residential environments. *Urban Forestry & Urban Greening*, 14(2):404–412, 2015.
- [37] John W. Lisle, Richard J., Brabham, Peter, Barnes. *Basic Geological Mapping*. 2011.
- [38] Manolis I. A. Lourakis and Antonis A. Argyros. SBA. *ACM Transactions on Mathematical Software*, 36(1):1–30, 2009.
- [39] D. G. Lowe. Distinctive image features from scale invariant keypoints. *Int'l Journal of Computer Vision*, 60(2):91–110, 2004.
- [40] D.G. Lowe. Object Recognition from Local Scale-Invariant Features. *IEEE International Conference on Computer Vision*, 1999.
- [41] Jason Mah, Claire Samson, and Stephen D. McKinnon. 3D laser imaging for joint orientation analysis. *International Journal of Rock Mechanics and Mining Sciences*, 48(6):932–941, 2011.
- [42] Jason Mah, Claire Samson, Stephen D. McKinnon, and Denis Thibodeau. 3D laser imaging for surface roughness analysis. *International Journal of Rock Mechanics and Mining Sciences*, 58:111–117, 2013.

- [43] T McLeod, C Samson, M Labrie, K Shehata, J Mah, P Lai, L Wang, and J H Elder. Using video acquired from an unmanned aerial vehicle (UAV) to measure fracture orientation in an open-pit mine. *Geomatica*, 67(3):173–180, 2013.
- [44] Tara McLeod. *Three-dimensional imaging applications in Earth Sciences using video data acquired from an unmanned aerial vehicle*. Doctoral dissertation, Carleton University, 2012.
- [45] M. Pauly, M. Gross, and L.P. Kobbelt. Efficient simplification of point-sampled surfaces. *IEEE Computer Society Proceedings of the conference on Visualization'02*, pages 163–170, 2002.
- [46] R M Post, J M Kemeny, and R Murphy. Image processing for automatic extraction of rock joint orientation data from digital images a forward. *Rock Mechanics in the National Interest*, 2001.
- [47] Jingbang Qiu, Ying Lu, Tianci Huang, and Takeshi Ikenaga. A FPGA-based real-time hardware accelerator for orientation calculation part in SIFT. *IIIH-MSP 2009 - 2009 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 1334–1337, 2009.
- [48] T Rabbani, F a van den Heuvel, and G Vosselman. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):248–253, 2006.
- [49] Cumhuri Sahin. Planar segmentation of indoor terrestrial laser scanning point clouds via distance function from a point to a plane. *Optics and Lasers in Engineering*, 64:23–31, 2015.
- [50] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2007.
- [51] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*, 1:519–528, 2006.
- [52] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics*, 25(3):835–846, 2006.

- [53] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian Surface Editing. *Eurographics Symposium on Geometry Processing*, pages 175–184, 2004.
- [54] G.W. Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.
- [55] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle Adjustment A Modern Synthesis Vision Algorithms: Theory and Practice. *Vision Algorithms: Theory and Practice*, 1883:153–177, 2000.
- [56] Darren Turner, Arko Lucieer, and Luke Wallace. Direct georeferencing of ultrahigh-resolution UAV imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2738–2745, 2014.
- [57] Darren Turner, Arko Lucieer, and Christopher Watson. An automated technique for generating georectified mosaics from ultra-high resolution Unmanned Aerial Vehicle (UAV) imagery, based on Structure from Motion (SFM) point clouds. *Remote Sensing*, 4(5):1392–1410, 2012.
- [58] Eric Turner and Avidesh Zakhor. Watertight planar surface meshing of indoor point-clouds with voxel carving. *Proceedings of the 2013 IEEE International Conference on 3D Vision, 3DV 2013*, pages 41–48, 2013.
- [59] Yathunanthan Vasuki, Eun Jung Holden, Peter Kovese, and Steven Micklethwaite. Semi-automatic mapping of geological Structures using UAV-based photogrammetric data: An image analysis approach. *Computers and Geosciences*, 69:22–32, 2014.
- [60] George Vosselman, B G H Gorte, G Sithole, and Tahir Rabbani. Recognising structure in laser scanner point clouds. *Remote Sensing and Spatial Information Sciences*, pages 33–38, 2004.
- [61] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds. ‘Structure-from-Motion’ photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012.
- [62] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. *Computer Vision and Pattern Recognition (CVPR)*, pages 3057–3064, 2011.
- [63] Changchang Wu. Towards linear-time incremental structure from motion. *Proceedings - 2013 International Conference on 3D Vision, 3DV 2013*, pages 127–134, 2013.

- [64] Yongjian Xi Yongjian Xi, Ye Duan Ye Duan, and Hongkai Zhao Hongkai Zhao. A non-parametric approach for noisy point data preprocessing. *2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics*, 1(2):2–7, 2009.

Curriculum Vitae

Name: Jonathan Kissi-Ameyaw

Education:	University of Central Lancashire	2002
	Manchester Metropolitan University	2005
	Open University	2009 - 2010
Qualifications:	BA(HONS) Online Journalism	2002
	MSc Computing	2005
	Short Courses - Cosmology/Mathematics	2009 - 2010

Honours and Awards:

Outstanding Presentation in Graduate Symposium for Robotics and Control	University of Western Ontario	2016
---	-------------------------------	------

Research Experience:

Graduate Research Assistant (MEdSc) <i>Supervisors: Dr K. McIsaac, Dr X. Wang</i>	University of Western Ontario	2014 - 2016
--	-------------------------------	-------------

Publications:

Manaud, N., et al.,	<i>The ESA Planetary Science Archive (PSA)</i>	2009
	<i>European Planetary Science Congress</i>	
	<i>2009. Vol. 1. 2009.</i>	
J. Kissi-Ameyaw, E. P. Monaghan, B. H. Foing	<i>Astronomy from the Moon: Possible Science Investigations and Precursors</i>	2008
	<i>LPI Contributions 1446 (2008): 73</i>	