

FRONTIER S
I >



Queensland
Government

Upgrading the Spatial Accuracy of the Digital Cadastral – A Pilot Study

October 2018

Project Details:

Project Organisations	CRCSI and DNRME
Project Number	2.32
Project Leaders	Russell Priebbenow (DNRME) Clive Fraser (CRCSI)
Principal Researcher	Sudarshan Karki (DNRME)
Report Preparation	Sudarshan Karki (DNRME)
Project Start Date	15 May 2017
Project Completion Date	29 October 2018
Project Aim	Evaluate the feasibility of utilising Lidar and Imagery to extract fence-lines for geo-positional upgrade of digital cadastre and evaluate the accuracies obtained
Project Objectives	<ul style="list-style-type: none"> ● Automate feature extraction methodologies and assess applicability for operational implementation ● Evaluate the contribution of Imagery and Lidar to upgrading the spatial accuracy of digital cadastre ● Identify accuracy achievable through these methods ● Recommendations on application of imagery and Lidar for digital cadastre spatial accuracy upgrade process

Project Resources:

Company	Services	Resource Persons
ESRI Australia	Consulting Services	Dipak Paudyal
Harris Geospatial Solutions Inc. USA	Processing Pipeline and GUI Development	Atle Borsholm Bryan Justice
RPS Australia	Lidar and Imagery Capture	Rajdeep Amatya

Acknowledgement

I would like to thank CRCSI for providing the funding that assisted in the successful completion of this project. I would also take this opportunity to thank my employer DNRME for jointly organising this project with CRCSI.

I am extremely grateful to our joint project leaders Dr Russell Priebbenow (DNRME) and Prof. Clive Fraser (CRCSI), as well as Phil Collier (CRCSI), for their constant guidance, supervision and feedbacks on technical aspects of the project and also for providing managerial and logistical support throughout the various stages of the project. Their support has been vital to deliver such a comprehensive project with many moving parts.

I would like to thank Esri Australia, Harris Geospatial (USA) and RPS Australia for their respective involvement and collaboration in various capacities during this project. In particular, I would like to thank Dr Dipak Paudyal from Esri Australia for providing technical advice at various stages of the project and coordinating the effort with Harris Geospatial. Special thanks go to Atle Borsholm and Bryan Justice from Harris Geospatial for the development of the Fence Extraction software using IDL. I would also like to thank Rajdeep Amatya from RPS for assisting with timely Lidar and Aerial Photo capture, vital for the execution of this project.

I would like to acknowledge the assistance I received from Garry Cislowski (DNRME) for organising a GPS field survey including the equipment, logistics, data transfer and processing. I would also like to thank Brittany Smith who provided valuable administrative support during the execution of this project; and Govinda Baral (DNRME) for the initial work that he did on the project. I would also like to thank Mike Burdett and Luke Tonkin (South Australia) for providing data from Adelaide for testing in this project.

I would like to extend my sincere thanks to all other individuals that have contributed to the success of this project. This project would not have been successful without the help and support that was received during the entire process that lasted almost a year, from project inception, through to data acquisition, processing and analysis, software development, testing and validation through to final delivery of the project.

Table of Contents

1	Introduction	1
1.1	Problem Statement	2
1.2	Project Aim	3
1.3	Project Objectives	3
1.4	Research Questions	3
1.5	Project Significance	4
1.6	Benefits of a Successful Outcome	5
1.7	Project Deliverables	5
1.8	Project Risks	5
2	Review of Previous Work	6
2.1	Image Processing	6
2.2	Lidar Processing	10
3	Study Area and Data Acquisition	19
3.1	Background	19
3.2	Terrain characteristics	19
3.3	Study area	20
3.3.1	Semi-urban test area: Morayfield	20
3.3.2	Rural test area: Toowoomba	20
3.4	Existing Data	21
3.5	Data acquisition	23
3.6	GPS Field survey	24
3.7	Image Ortho-rectification	26
4	Methodology	29
4.1	Background	29
4.2	Research Steps	30
4.2.1	Project set-up, Literature Review and Planning	30
4.2.2	Accuracy requirements for Queensland cadastre	30
4.2.3	Study area identification	31
4.2.4	Manual feature extraction and accuracy analysis	31

4.2.5	Evaluation of alternative methods	32
4.2.6	Processing pipeline and software development plan	32
4.2.7	Automated feature extraction and data analysis	33
4.2.8	Evaluation of feature extraction	33
4.2.9	Reporting and recommendations:	33
4.3	Methods of Feature Extraction Explored	34
4.3.1	Overview of Methods Explored	34
4.3.2	Method using Power-line Extraction Parameters	35
4.3.3	Lidar Surface Difference Method	37
4.3.4	Lidar Surface Difference on Imagery Output	38
4.3.5	Lastools Direct Height Filter Method	38
4.3.6	Image Edge-Detection Method	39
4.3.7	Image-derived Point-Cloud Filter Method	41
4.3.8	Image Segmentation Method	42
4.4	Considerations for Fences and Corresponding Algorithms	44
4.5	Workflow developed for Fence-line detection using Lidar	46
4.6	Workflow for Fence-line detection using Integrated Imagery and Lidar	48
4.7	Summary	50
5	Data Processing and Evaluation of DCDB Upgrade	51
5.1	Background	51
5.2	Fence-line Extraction Algorithm Development	51
5.2.1	Algorithm Processing Stages – Lidar	51
5.2.2	Fence-line detection using Integrated Imagery and Lidar	55
5.2.3	Quality Assessment Tool for Fence-line extraction	58
5.2.4	SVM implementation in ENVI	61
5.2.5	SVM implementation in the GUI	62
5.2.6	Implementation Versions of the GUI	63
5.3	Assessment of Algorithm for Fence-line Extraction	64
5.3.1	Assessing the algorithm and data for rural areas	64
5.3.2	Assessing the feature extraction for Geiger Mode Lidar	67
5.3.3	Assessing the algorithm and data for Adelaide	68

5.4	Assessment of feasibility of DCDB upgrade using Fence-lines	70
5.5	Assessment of Various Lidar Data Resolutions for Fence-lines	73
5.5.1	Evaluation of high resolution Lidar	73
5.5.2	Evaluation of medium resolution Lidar	74
5.5.3	Evaluation of low resolution Lidar	76
5.6	Summary	78
6	Accuracy Achievable	79
6.1	Accuracy of Lidar Data	79
6.2	Accuracy of Filtering using SVM	80
6.3	Accuracy of Extracted Fence-lines	82
6.4	Sources of Error in Fence-line extraction	85
6.4.1	Missing Fence-lines	85
6.4.2	Inaccurate Line Direction	87
6.4.3	False Positives	87
6.5	Summary	89
7	Recommendation for Cadastral Upgrade	90
7.1	Background	90
7.2	Operational Opportunities	90
7.2.1	Potential Benefits to Organisations	90
7.2.2	Time savings	91
7.2.3	Accuracy Attained	91
7.2.4	Process Integration	91
7.3	Limitations	92
7.3.1	Accuracy Expectations	92
7.3.2	Time Expectations	92
7.3.3	Ease of Adaptation of the Algorithm	92
7.3.4	Data Availability	93
7.3.5	Existing processes	93
7.4	Commercialisation Opportunities for CRCSI	94
7.5	Processing Times for Fence-line extraction	94
7.6	Lidar and Imagery Capture – Relative Advantages	96

7.7 Lidar and Imagery – Recommended Capture Resolution	97
7.8 Algorithm Improvement	97
7.9 Summary	97
8 Conclusion and Future Research	98
8.1 Review of Aim and Objectives	98
8.2 Objective 1: Upgrade Methodologies	99
8.3 Objective 2: Evaluation of Data Sources	100
8.4 Objective 3: Accuracy Achievable	100
8.5 Objective 4: Recommendations	102
8.6 Future Research	102
9 References	104
10 Appendices	110
10.1 Positional Accuracy of DCDB in Queensland	110
10.2 Camera Calibration Report Extract	111
10.3 Photos Lidar + Imagery Data Capture	113
10.4 Photos GPS Field Survey	115
10.5 Processing in Lastools	116
10.6 Input Parameters for the Developed GUI	120
10.7 How to Process Lidar data in the Developed GUI	125
10.8 Project Budget and Expenditure	129
10.9 Project Timeline	130

List of Tables

Table 2-1: New Filters or algorithms developed in these papers	13
Table 2-2: Studies with feature extraction using Image and Lidar fusion method	17
Table 3-1: Extract from AUSPOS Solution for GPS Field Survey Calculations	25
Table 4-1: Fence-line characteristics and algorithm considerations	44
Table 4-2: Lidar Density vs. parameters to be used	46
Table 5-1: Algorithm parameters specific to fence extraction	53
Table 6-1: Lidar data validation using GPS field survey coordinates	79
Table 6-2: Confusion Matrix from SVM Training	81
Table 6-3: Classification metrics from the Confusion Matrix	81
Table 6-4: Computing the accuracy of Fence-line extraction from Lidar (Top and Bottom table)	83
Table 6-5: Computing the accuracy of Fence-line extracted from Imagery & Lidar (Top and Bottom table)	83
Table 7-1: Time taken for fence-line extraction from various Lidar data sources and resolutions	95
Table 7-2: Time calculation for fence-line extraction in another computer at Harris Geospatial, USA	95

List of Figures

Figure 3-1: (a) Extent of the Project area in Morayfield, and (b) Areal image of the Project area	20
Figure 3-2: (a) Extent of the Project area, (b) Areal image of the Project area	21
Figure 3-3: (Left) Photo centre over Morayfield for aerial imagery data capture, (Right) Flight lines for Lidar and Imagery data capture	23
Figure 3-4: GPS Field Survey locations	24
Figure 3-5: Image capture and processing (Top) Aerial photo of 6cm GSD; (Middle) Systematic shift noticed during ortho-rectification; (Bottom) Ortho-rectified imagery using GPS ground control	27
Figure 4-1: Processing overview for (Top) Lidar data and (Bottom) Integrated Imagery and Lidar data	32
Figure 4-2: Various processes explored for fence-line extraction	34
Figure 4-3: Workflow for Powerline Extraction Algorithm	35
Figure 4-4: Points in white represent fence but other unclassified points are classes as fence as well	36
Figure 4-5: Fences detected using powerline algorithm showing false positives and omissions	36
Figure 4-6: Surface difference from DSM-DEM derived from Lidar	37
Figure 4-7: (Left) Lidar surface difference model, and (Right) Extracted fence-lines on imagery	38
Figure 4-8: Fence-line raster filtered using Lastools	39
Figure 4-9: (Top) Edge detection shows a large number of edges; and (Bottom) Buildings overlaid on the edges	40
Figure 4-10: Dense point cloud generated from stereo-pair images by image matching	41
Figure 4-11: Feature extraction from aerial image derived point cloud	42
Figure 4-12: Image segmentation rule creation window with moving overview window	43
Figure 4-13: Fence-line extracted using image segmentation with false positives and omissions	43
Figure 4-14: Workflow for Fence-line extraction using Lidar data	47
Figure 4-15: Workflow for Fence-line extraction using Ortho-imagery and Lidar Relative Elevation	49
Figure 5-1: Trees causing many edges, but no straight line edges	55
Figure 5-2: Roof texture causing many edges	56
Figure 5-3: Shadows and roads are detected in this case	56
Figure 5-4: While the fence on the left is correctly identified, the algorithm also finds many other lines on the building	56
Figure 5-5: More buildings as well as some fence lines are detected here	57
Figure 5-6: SVM filtering results in lines in green to be selected as fences, and lines in red are rejected	57
Figure 5-7: Various options explored for background imagery to be used in visual quality assessment, manual editing and SVM training	58
Figure 5-8: (Left) Aerial imagery background for non-ENVI version and (Right) fence-line overlay	59

Figure 5-9: (Left) Probability values shown for each line segment; and (Right) SVM training model creation and accuracy reporting	60
Figure 5-10: Example fence image in rural areas	65
Figure 5-11: (Left) Lidar on fences and (Right) fence-line extraction	65
Figure 5-12: Example of rural fence-line extraction and SVM filtering	66
Figure 5-13: (Left) Study Area in USA; and (Right) Geiger Mode Lidar point cloud	67
Figure 5-14: (Left) Unfiltered fence-lines and (Right) Filtered fence-lines extracted from Geiger Mode Lidar	67
Figure 5-15: Adelaide Lidar processing (Top) Unfiltered fence-lines (Bottom) Filtered lines	68
Figure 5-16: DCDB metadata states high accuracy but Lidar fence-lines match imagery not DCDB	69
Figure 5-17: DCDB metadata states low accuracy, still Lidar fence-line extract matches imagery not DCDB	70
Figure 5-18: Simulated distortion of the digital cadastre	71
Figure 5-19: Link lines between extracted fence-lines and distorted cadastre for rubber-sheeting	71
Figure 5-20: Variable distance and direction of the link lines due to rotation, translation and scaling of the distorted cadastre	72
Figure 5-21: The DCDB was able to be block adjusted to its original high accurate position using the extracted fence-lines	72
Figure 5-22: Assessment of various Lidar resolutions and capture directions for fence-line extraction	73
Figure 5-23: Fence-line extraction from high-resolution extraction shows a large percentage of visible fence-lines extracted	74
Figure 5-24: Fence-lines extracted from medium resolution Lidar for (Top) Horizontal and (Middle) Vertical directions of flight; and (Bottom) the combined Lidar results	76
Figure 5-25: Fence-lines extracted from low resolution Lidar for (Top) Horizontal and (Middle) Vertical directions of flight; and (Bottom) the combined Lidar results	78
Figure 6-1: Example SVM Training Window	80
Figure 6-2: Small building/shed (blue X) is causing a section of the fence to be missed	85
Figure 6-3: An example of fence line segments missed due to vegetation	86
Figure 6-4: Missing fence intersection lines because of circular coverage reduction of point cluster by the tree removal algorithm	86
Figure 6-5: Fence-line being skewed towards a cluster of vegetation	87
Figure 6-6: An example of parked cars being misclassified as a fence line is shown here, (blue arrow)	88
Figure 6-7: An example of the sides of a building being misclassified as a fence line (not common in Lidar extraction but very common in image-based extraction)	88

List of Abbreviations

CORS	Continuously Operating Reference Station
CRC-SI	Collaborative Research Centre - Spatial Information
DCDB	Digital Cadastral Database
DEM	Digital Elevation Model
DNRME	Department of Natural Resources, Mines and Energy
DSM	Digital Surface Model
ENVI	Environment for Visualizing Images
ESRI	Environmental Systems Research Institute
GSD	Ground Sampling Distance
GUI	Graphical User Interface
IDL	Interactive Data Language
JSON	JavaScript Object Notation
Lidar	Light Detection and Ranging
PPSM	Points Per Square Metre
PU	Positional Uncertainty
RANSAC	Random Sample Consensus
RBF	Radial Basis Function
RMSE	Root Mean Square Error
SVM	Support Vector Machine

Summary:

Background: The digital cadastre is one of the most fundamental spatial layers that are used by various organisations, however, the geo-positional accuracy of the digital cadastre is variable which causes numerous problems. The traditional costs of upgrading the spatial accuracy is very time-consuming and costly. This project explores the feasibility of semi-automatic or automatic feature extraction of fence-lines using Lidar and Imagery as well as evaluate the feasibility of using these fence-lines for upgrading the digital cadastre and the accuracies obtained.

Objectives: The objectives of this project is to evaluate the suitability of imagery and Lidar for upgrading the spatial accuracy of digital cadastre; assess the accuracy achievable through these methods; develop (semi-) automatic feature extraction methodologies and assess applicability for operational implementation; and provide recommendations on application of imagery and Lidar for the upgrade of the digital cadastre.

Data: As the developed methodology was expected to operate on both urban and rural areas, Lidar data (2ppsm) and Imagery (10cm GSD) was available for a rural area near Toowoomba and for a semi-urban area at Morayfield both in Queensland. The data for Morayfield was captured over two different times and had different combined point density (24 ppsm and 64 ppsm), while the ground sampling distance for the imagery were 10cm and 6 cm respectively. The Morayfield Lidar data was verified using differential GPS field survey which demonstrated that the RMS error for vertical accuracy of the Lidar data ranged between 3mm to 3cm. The GPS field survey coordinates was further used for image rectification. Digital cadastral data was available for both the areas. Further Lidar, imagery and cadastral data was made available for Adelaide, South Australia for testing of the algorithm in an urban area with different fence-line characteristics.

Methodology: A multi-pronged approach in feature extraction was used to evaluate the suitability of imagery and Lidar for extracting linear features (fence-lines) with a view to upgrading the spatial accuracy of digital cadastre. A critical part of the development effort was to ensure that the method was suitable to cater to a variety of data sources under wide ranging environment for the capture and availability of data sources. It was also considered critical for development of a GUI with a user-friendly framework for quality assessment and refinements of the result. The method was designed to automate a majority of the steps in the selected workflows. After initial exploration of multiple feature extraction methods, it was identified that Lidar based methods (under sparse, medium and high-density point cloud collection conditions) provided more robust feature extraction results using automated methods and thus remained the core focus for the remainder of the project work. To cater for areas which lacked Lidar point clouds or benefited from existing imagery, additional Image-based feature extraction methods were developed to complement the Lidar-based fence-line extraction methods. The project methodology was thus designed to account for use of Lidar point clouds collected at various point densities and aerial photos at multiple resolutions. Design of the methodology also catered for semi-urban and rural areas that accounted for physical differences in actual fence features. Finally, a GUI was developed with consideration given to the provision of a software that was robust and easy to use either on an ENVI/IDL or an open source environment.

Results: The developed workflow has shown promising results, with extraction accuracy that should allow for an accurate adjustment of the existing cadastre. Fence-lines extracted from Lidar have a combined horizontal accuracy of 0.282m while fence-lines extracted from imagery have a combined horizontal accuracy of 0.258m. Although the accuracy of image-based feature extraction appears to be better, the number of fence-line segments extracted from Lidar is significantly higher in number.

Conclusion: This aim of the project was to explore the feasibility of using Lidar point cloud and imagery data for feature extraction leading to improvement of existing cadastral survey using fence-lines. This goal was achieved by thorough testing and development of a workflow for semi-automated extraction of fence-line boundaries from airborne laser scanning and imagery data. To ensure that the developed workflow could be implemented over rural and urban areas, the study areas were chosen with such characteristics and with varying data density to simulate real conditions. Thus, it can be concluded that Lidar based fence-line extraction can be used, and where necessary augmented by image-based extraction using the methods and workflows developed during this project and cadastral boundaries block adjusted with a very high degree of confidence.

1 Introduction

The Digital Cadastral Database (DCDB) is the spatial representation of current land parcels in a jurisdiction. It usually includes a legal Lot on Plan description and relevant attributes and a graphical representation of the parcels. The DCDB provides the map base for systems dealing with land related information.

The DCDB in Queensland was created by digitising existing cadastral maps at a variety of scales and accuracies. A positional accuracy value has been allocated to all parcels in the DCDB. The value reflects the maximum error status of the parcel and has been derived from the capture process or assigned as the spatial accuracy of the DCDB has been upgraded. The maximum error in the DCDB is currently +/- 63m. The maximum error status is based on an assumed plotting accuracy for the source mapping.

Stakeholders rely on the graphical representation of the parcels to be spatially accurate. Since the initial capture of the DCDB from paper based maps, data users are now utilising the data in ways that requires a higher spatial accuracy than was achieved in the initial capture. The difference in the position of the DCDB is visible to users when overlaid over aerial and satellite imagery. Councils, utilities and other agencies involved in asset management now have highly spatially accurate data of their assets which they attempt to link to the DCDB.

This project explores the extent to which Lidar data, in cases complemented by high-resolution aerial imagery can be used to upgrade the spatial accuracy of the digital cadastre. Automated and semi-automated feature extraction are employed to detect, extract and validate the location of natural and man-made features which may indicate the location of property boundaries, and these are correlated with existing digital cadastral data in order to identify and rectify geo-positional biases within the existing digital representation of the cadastre.

The primary focus of the project is upon enhancement of the absolute accuracy of the cadastral database rather than upon its relative accuracy, though overall spatial accuracy upgrading in non-urban areas are also addressed. The project centres upon two main research components.

The first involves the selection and evaluation of candidate 2D and 3D feature extraction tools. The second comprises a pilot study, largely empirical in nature, which involves the determination of geo-positional biases between the digital cadastre and land parcel boundary segments extracted via the feature extraction approach.

The project culminates in this report that not only presents the results of the pilot study and assesses the relative value of various data sources and feature extraction methodologies, but also identifies operational opportunities and limitations.

1.1 Problem Statement

The digital cadastre is one of the most fundamental and important spatial layers maintained by land agencies across Australia and NZ. It is used by a multitude of “downstream” organisations in a diverse range of applications and provides the frame of reference for a number of other spatial layers.

Notwithstanding its critical role, the digital cadastre is, in many ways, not fit-for-purpose on account of its variable spatial accuracy (Appendix 10.1). The time and cost of upgrading the accuracy of the digital cadastre by field survey procedures is beyond what the relevant agencies can realistically afford. An alternative methodology is required that brings substantial cost and productivity gains, while delivering a worthwhile improvement in spatial accuracy, especially absolute geo-positional accuracy.

Without exception, efficiency and productivity gains come at a cost. In the case of upgrading the digital cadastre using remote sensing data, there is the potential in certain situations of a managed accuracy cost being incurred. The question then is: will this alternative remote sensing approach deliver sufficient accuracy at a reasonable cost and in a reasonable timeframe? This is the question to be answered by this project.

The research project will in particular, investigate the use of various remotely sensed data, namely imagery and Lidar, and evaluate their suitability in the context of upgrading the spatial accuracy of the cadastre.

The intent is to not only provide an empirical assessment of the alternative technology options, but to make recommendations on how and under what conditions, the findings of the project might be applied in an operational setting by DNRME and, potentially, by other land agencies across Australia and NZ.

1.2 Project Aim

The aim of this project is to evaluate the feasibility of utilising Lidar and Imagery to extract fence-lines for geo-positional upgrade of digital cadastre and evaluate the accuracies obtained.

1.3 Project Objectives

1. To develop upgrading methodologies for cadastral data based on automated feature extraction and to assess their applicability and potential for operational implementation by partner land agencies;
2. To evaluate the contribution of remotely sensed data sources (e.g. airborne and satellite imagery, and Lidar) to upgrading the spatial accuracy of the digital cadastre;
3. To identify, through experimental testing, the accuracy achievable from those data sources individually and in combination;
4. To deliver recommendations on how and under what conditions remote sensing data might be employed for cadastral upgrade purposes.

1.4 Research Questions

The following research questions are formulated which will assist to achieve the aim and objectives of this project:

1. Is it possible to detect and extract, both manually and automatically, geo-positional data using either Lidar data or Ortho-imagery complemented by Lidar data, to accuracy levels corresponding to those specified for digital cadastral data?

2. What spatial resolutions and metric quality of input data are required to meet cadastral accuracy requirements; i.e. what density of Lidar data and what sources and scales of imagery (airborne and satellite) are appropriate to support boundary feature extraction for cadastral upgrade?
3. What developments in, and implementations of 2D and 3D automated feature extraction are necessary to support semi- and fully-automatic workflows and data processing pipelines for the process of upgrading the digital cadastre from imagery?
4. Can the georeferencing of feature data extracted via the developed methodology be employed to effect an upgrading of the current cadastre to the required accuracy, primarily the absolute geo-positioning accuracy of the digital cadastre?
5. Does the proposed scenario for upgrading the accuracy of the digital cadastre via imagery and Lidar data have potential in the future for operational implementation by the relevant custodians of the cadastre?

1.5 Project Significance

The output of the project includes procedures and associated processing workflows, graphical user interface and related source codes and computational tools for Lidar and integrated image-based extraction of fence-lines for upgrading the cadastre.

It provides a new and more efficient means to maintain geo-positional accuracy of the digital cadastre and is expected to automate aspects of the upgrade process which would otherwise be an expensive and relatively slow manual process.

There is potential for significant scientific or technical impact through process automation and a new approach to cadastral data acquisition and upgrading as well as the exploration of new methods to extract features that are extremely narrow and have associated elevation ranges.

The target of this project is land agencies who are custodians of the cadastre. The processes developed in this research provide an alternative approach to the current, labour-intensive and expensive land survey or image based technique for cadastral upgrading.

1.6 Benefits of a Successful Outcome

The project demonstrates the benefits of CRCSI R&D expertise in the enhancement/upgrading of both the accuracy and operational work processes related to an important spatial data set, namely the cadastre.

The success of this project demonstrates the capability of the CRCSI and partner agencies to work together in the conduct and management of applied research that aims to enhance the business capabilities and products of Government sector participants.

1.7 Project Deliverables

The preparation of a report that not only presents the results of the pilot study and assesses the relative value of various data sources and automated feature extraction/upgrading methodologies, but also identifies operational opportunities and limitations, and draws conclusions regarding the prospects for adoption of the developed methodology by the partner agencies. Software tools, algorithms, workflow documentation, etc. are also made available to partners for either adoption or further development.

1.8 Project Risks

The accuracy of image-based cadastral data collection falls short of requirements: To mitigate that it is necessary to ensure comprehensive initial analysis of data sources & take account of differing requirements between urban and non-urban cadastres.

Development of automated process, while demonstrating concept feasibility, may fall short of operational requirements & thus not be implemented by DNRME or other land agencies: To mitigate that it suggested to ensure both developed manual and semi-automated workflows offer practical alternatives in such circumstances.

2 Review of Previous Work

This section reviews some of the previous work done in Image processing and Lidar processing relevant to this project which is feature extraction for linear features that would assist in the extraction of fence-lines for cadastral data geo-positional upgrade.

This section explores some of the work done in image processing first, followed by Lidar processing work, which then leads to a combined Lidar and Imagery processing field of research.

2.1 Image Processing

There has been a relatively large body of work for detection of linear/rectilinear features from LIDAR Point cloud and Imagery with varying level of success in feature extraction. While some of these extraction methods may be applied to update the cadastre, no literature were found in the scope of the search for this project discussing the extraction of fence-lines for updating the cadastre.

The focus of this review is therefore to explore the availability of methods and techniques that are used for generalized detection of linear features that could then be extrapolated to detection of features like fences. The literature discuss feature extraction methods using mainly two data sets, namely Imagery and Lidar point cloud. The third approach or method is fusing the information derived from Imagery and LIDAR as a combined method.

Image based methods can broadly described as edge detection methods using one or the other type of edge detectors. One of the most popular method for Image based detector is Canny Edge detector (Canny 1986; Ding and Goshtasby 2001; Green 2002; Juneja and Sandhu 2009; Biswas and Sil 2012; Shrivakshan and Chandrasekar 2012).

The edge detectors of images usually are of the following type:

Gradient edge detectors - detects the edges by looking for the maximum and minimum in the first derivative of the image (Shrivakshan and Chandrasekar 2012); uses first directional derivative operation and includes algorithms such as Sobel operator (Sharifi *et al.* 2002);

Laplacian - The Laplacian (Mexican Hat operator) method searches for the zero crossings in the second derivative of the image to find edges; (Shrivakshan and Chandrasekar 2012);

Zero Crossing – uses second derivative and includes Laplacian operator and second directional derivative; (Sharifi *et al.* 2002);

Laplacian of Gaussian – developed by (Marr and Hildreth 1980) as a combination of Gaussian filtering with the Laplacian;

Gaussian edge detectors – symmetric along the edge and reduces the noise by smoothing the image; includes Canny edge detector;

Coloured edge detectors – divided into three categories of output; Fusion methods, Multi-dimensional gradient methods and Vector methods.

Linear Methods - Also known as linear method, it involves discrete approximation of the first order derivative in a given direction, Pratt (0° , 45° , 90° , 135°); Prewitt (compass direction), Argyle's operator (combination with Gaussian); Macleod's operator (combination with Gaussian) (Peli and Malah 1982);

Non-Linear Methods – 2×2 or 3×3 window, gradient which is defined as the maximum over Θ_n of the magnitude of the partial derivative in direction Θ_n . Roberts: $\max\{|f(i, j) - f(i + 1, j + 1)|, |f(i, j + 1) - f(i + 1, j)|\}$; Sobel: (3×3 window); Prewitt: (same operator as Sobel, different scaling); Kirsch; and Robinson: (3-level and 5-level, template matching using a set of masks to determine the existence of an edge and direction); Abdou (extended the masks to larger window sizes 5×5 , 7×7 , 9×9); Wallis (Laplacian on the logarithm of the intensity); Smith and Davis (two operators for binary and grey level images that measure the ratio between the balance of a bi-modal distribution and a measure of disorder); Hale (implement a two-dimensional operator by rotating a one-dimensional operator); Rosenfield (computing

differences between averages of non-overlapping neighbourhoods that meet at the same point); (Peli and Malah 1982)

Best Fit Method – best fit of a function to a given image, Hueckel's method (computed parameters based on orthogonal functions from comparison of circle and ideal 2D edge); Abdou (optimal edge fitting based on discrete image model); Modestino and Fries (determine filter operation such that its operation on a noisy image is the best approximation to the operation of Laplacian on the ideal image); (Peli and Malah 1982).

Several authors have developed performance evaluation criteria for edge detectors:

Precision, Resolution and Accuracy (Sharifi *et al.* 2002); Signal to Noise Ratio (SNR) and Average Risk defined as the ratio of the number of detected edge points which do not coincide with the ideal edge, to the number of detected edge points which coincide with the ideal edge (Peli and Malah 1982); Error rate, Localisation and Response by Canny; Distribution of the detected true edge points, developed by Fram and Deutsch as well as Maximum likelihood estimate of the ratio of the total number of true edge points to the total number of detected edge points; Weighted and normalised deviation of the real edge from the ideal edge line, developed by Pratt;

Further quantitative performance evaluation criteria were: Percentage of edge points detected on the ideal (desired) edge; Number of detected edge points which do not coincide with the ideal edge (normalised by the number of points on the edge); Mean width of a detected edge, defined as the ratio of the total number of detected edge points to the number of ideal edge points; Weighted and normalised deviation of an actual edge point from the ideal edge as defined by Pratt; Average squared deviation of a detected edge point from the ideal edge; and Mean absolute value of deviation (Peli and Malah 1982).

Additionally, qualitative edge detection performance criteria were: Type of contour (perfect edge, broken edge, perfect but broken at critical points such as corner of square); Single or double edge (a single or two separate edges); and Distortion (shift of the edge) (Peli and Malah 1982).

Edge detectors are expected to have the following properties (Ziou and Tabbone 1998):

Detect properties of image such as discontinuities in the photometrical, geometrical and physical characteristics. Variations in the grey level of the image include discontinuities (step edges), local extrema (line edges) and edge meets (junctions). Physical edges correspond to variations in reflectance, illumination, orientation, and depth of surfaces and is proportional to scene radiance which is represented in the images by the change in intensity function.

Smoothing of Image – positive effect – reduce noise, ensure robust edge detection; negative effect – information loss;

Image Differentiation – is the computation of the necessary derivatives to localise the edges (localise variations of the image grey level and to identify the physical phenomena that produced them). The differentiation operator is characterised by its order, its invariance to rotation and its linearity;

Edge Labelling – localising edges and increasing signal-to-noise ratio by suppressing false edges and involves finding the local maxima along the gradient vector;

Multi-detector and Multi-scale approaches – common convolution operators are of the form $(f_s * I)(x,y)$ where I is the image, f_s is the filter and s the scale; multiple edge detectors and scales are necessary for multiple images.

2.2 Lidar Processing

Data processing can be automatic or semi-automatic and can include multiple operations such as filtering, modelling of systematic errors, feature detection and line thinning (Sithole and Vosselman 2003b).

Most of the Lidar filtering methods facilitate the extraction of features of interest by filtering the element that would potentially confuse an accurate extraction process by modelling an accurate bare earth surface that becomes the backbone of downstream processes. These methods include filters such as:

Iterative linear least squares interpolation – which removes a low-degree polynomial trend surface from the original elevation data to produce a set of reduced elevation values. Here, firstly a rough approximation of the terrain surface is created, then sign of the residual checked which says negative values are terrain and the process is iterated (Liu 2008).

Comparative local curvature filter – used to filter tree points by comparing local curvatures of point measurements which was developed by (Haugerud and Harding 2001) and analysed in (Zhang *et al.* 2003)

Adaptive TIN model – used to find ground points in urban areas. Firstly, seed ground points within a user-defined grid of a size greater than the largest non-ground features are selected to compose an initial ground dataset. Then, one point above each TIN facet is added to the ground dataset every iteration if its parameters are below threshold values. The iteration continues until no points can be added to the ground dataset. However, the problem with the adaptive TIN method is that different thresholds have to be given for various land cover types. This method was proposed by Axelsson and analysed by (Zhang *et al.* 2003).

Slope based filter – identifies ground data by comparing slopes between a LIDAR point and its neighbours. A point is classified as a ground measurement if the maximum value of slopes between this point and any other point within a given circle is less than a predefined threshold. The lower the threshold slope, the more objects will be removed. The threshold slope for a certain area is either constant or a function of distance. A reasonable threshold slope can be

obtained by using prior knowledge about terrain in the study area. This method works well in flat urban areas, but has errors when applied to vegetated or variable slope areas. It assumes that the gradient of the natural slope of the terrain is distinctly different from non-terrain objects. This method was proposed by Vosselman and analysed by (Zhang *et al.* 2003);

$$DEM = p_i \in A \mid \forall p_j \in A : h_{p_i} - h_{p_j} \leq \Delta h_{max} (d(p_i, p_j)) \quad (\text{Vosselman 2000})$$

(Vosselman 2000) proposed three ways to derive filter kernels based on knowledge about the height differences in the terrain (filters 2 and 3 make use of training data set):

1. *Synthetic Function* – based on terrain shape and precision of height measurements:

$$\Delta h_{max} (d) = 0.3d + 1.65\sqrt{2}\sigma$$

Where 0.3 is for 30% terrain slope (different for different slope), and the second term is to allow that 5% of the terrain points with a standard deviation σ may be rejected

2. *Preserving important terrain features* – derive the terrain shape characteristics from a training sample consisting only of ground points such that the points in this area can be used to empirically derive the maximum height differences as a function of the distance between two points – this filter assists to maintain important terrain features but may also accept points that are non-ground:

$$f_{max}(\Delta h) = \frac{\partial F_{max}(\Delta h)}{\partial \Delta h} = N F(\Delta h)^{N-1} \frac{\partial F(\Delta h)}{\partial \Delta h} = N F(\Delta h)^{N-1} f(\Delta h)$$

3. *Minimising classification errors* – minimising errors by omission or commission for ground points. If the height of a ground point at a given distance is known, determine height differences between other points using probabilities derived from frequency counts of height differences between point pairs in a training data set of ground points, and between point pairs from the training set of ground points and other point from the set of unfiltered data. These values of height differences can be taken as the maximum height differences that are allowed in the filtered data in order to minimise the number of classification errors:

$$P(p_i \in DEM | \Delta h, d, p_j \in DEM) = \frac{P(p_i \in DEM, \Delta h, d, p_j \in DEM)}{P(\Delta h, d, p_j \in DEM)}$$

$$= \frac{P(\Delta h | d, p_i \in DEM, p_j \in DEM) P(p_i \in DEM | d, p_j \in DEM)}{P(\Delta h | d, p_j \in DEM)}$$

Mathematical morphological filter are a type of slope based filter mainly used for bare-earth extraction. Lidar images are converted to regular, greyscale, grid image in terms of elevation, while shapes of elevated buildings, cars etc. can be identified by change in grey tone, and algebraic set operations are performed to identify objects (Zhang *et al.* 2003). The main objective is to classify Lidar data into two classes, namely ground and non-ground points (Vosselman 2000; Aktaruzzaman and Schmitt 2010). Ground data is used for DTM generation while non-ground data is used for object detection and subsequent classification. Other results by various authors are as follows: large height difference is unlikely to be caused by steep slope in terrain (Vosselman 2000; Baligh *et al.* 2008); process of finding local minima and identifying terrain points from coarse to finer grid (Hu, Y. 2003); establishing the topological and geometric relations between bare-earth and surface objects, identifying surfaces whose perimeter is raised above the neighbourhood (Sithole and Vosselman 2003a).

Spectral Information Integration: Mapping spectral value from image pixel to Lidar point data mostly used for misclassified points between buildings and trees.

Calculate colour index: $CI = \text{green} / (\text{red} + \text{green} + \text{blue})$

where the index classified values are as: buildings $< 0.35 <$ trees. (Aktaruzzaman and Schmitt 2010)

A brief look at the literature was also made with a view to identify error in LIDAR data and its classification and for methods used to detect these errors: *Commission error* which results in classification of non-ground points as ground measurements (Vosselman 2000; Zhang *et al.* 2003); *Omission error* – removes ground points mistakenly (Zhang *et al.* 2003); and *Systematic error* which is visible when there are differences in height when combining data from adjacent

strips (Vosselman 2000) and can be eliminated by modelling the errors and performing a strip adjustment.

Methods of error detection include *Wavelet De-Noising* to assist in evaluating the response of filters (Baligh *et al.* 2008); *Ground truth; Derived filter functions* to check against other filter results (Vosselman 2000); and *Manual comparison* to evaluate filter performance against that performed with manual filtering (Sithole and Vosselman 2003b).

In a report submitted to ISPRS, (Sithole and Vosselman 2003b) make a comparative assessment of different filters. A comparison of open-source Lidar filtering algorithms in a forest environment is made by (Montealegre *et al.* 2015). An extensive review of filtering methods and algorithms and overview of LIDAR point cloud processing software is abundant in literature: (Tao and Hu 2001; Zhang and Whitman 2005; Fernandez *et al.* 2007; Baligh *et al.* 2008; Meng, Currit, and Zhao 2010).

Issues with computational efficiency are documented by (Sithole and Vosselman 2003a); a brief study on the Lidar data capture accuracy is discussed by (Montealegre *et al.* 2015) who generally stipulates this to be 0.15m in vertical and 1m horizontal.

New filters and algorithms for classification have been developed by several authors. Table 2-1 shows a list of methods and the authors who developed them that were researched for this project.

Table 2-1: New Filters or algorithms developed in these papers

Developed by	Method
(Zhang <i>et al.</i> 2003)	Progressive morphological filter for removing non-ground measurements from airborne LIDAR data
(Brunn and Weidner 1998)	Hierarchical Bayesian nets for building extraction using dense DSM
(Charaniya <i>et al.</i> 2004)	Supervised parametric classification of ALS
(Chen, C. <i>et al.</i> 2017)	Fast and robust interpolation filter for ALS point clouds

(Chen, Q. <i>et al.</i> 2007)	Filtering ALS data with morphological methods
(Silván-Cardenás and Wang 2006)	Multi-resolution approach for filtering Lidar altimetry data
(Vosselman 2000)	Slope based filtering of laser altimetry data
(Wang, O. <i>et al.</i> 2006)	Bayesian approach to building footprint extraction from ALS
(Aktaruzzaman and Schmitt 2010)	Automatic object detection to support urban flooding simulation
(Elmqvist <i>et al.</i> 2001), (Elmqvist 2002)	Active contours – applied to Lidar data the active shape model behaves like a membrane floating from underneath the data points
(Sohn, G and Dowman 2002)	Regularisation method – TIN progressively densified and points on TIN are bare-earth while the rest are objects
(Roggero 2001)	Modified slope based filter – variant of the morphological filter developed by Vosselman
(Brovelli <i>et al.</i> 2002)	Spline interpolation – made of five steps, Pre-processing; Edge detection; Region growing; Correction; and DTM computation
(Wack and Wimmer 2002)	Hierarchical modified block minimum – algorithm where DEMs of progressively lower resolutions are created
(Axelsson 1999, 2000)	Progressive TIN densification – a sparse to dense TIN is derived from Lidar points based on threshold values
(Sithole and Vosselman 2001)	Modified slope based filter – variant of morphological filter developed by Vosselman, works by pushing up vertically a structuring element (in the shape of an inverted bowl) from underneath a point cloud
(Pfeifer <i>et al.</i> 1999; Pfeifer <i>et al.</i> 2001), (Kraus and Pfeifer 1998, 2001), (Briese and Pfeifer 2001)	Hierarchical robust interpolation - a rough approximation of the terrain is first computed. The vertical distance of the points to this approximate surface is then used in a weight function to assign weights to all points. Points above the surface are given a small weight and those below the surface are given a large weight. In this way the recomputed surface is attracted to the low points.

An extensive review of literature was made to assess unique applications of LIDAR filters with a view to implement and/or improve techniques to be used for this project. These unique applications include Automatic object detection (Aktaruzzaman and Schmitt 2010), Automatic structure detection in a point cloud of an urban landscape (Sithole and Vosselman 2003a), Segmentation of unstructured point clouds (Bassier *et al.* 2017), Lidar data classification using extinction profiles and composite kernel (Ghamisi and Höfle 2017), Tensor modelling based ALS data classification (Li *et al.* 2016) and Hough-transform and other algorithms for automatic detection of 3D building roof planes from Lidar (Tarsha-Kurdi *et al.* 2007).

A review of the literature was made to understand current level of accuracy, performance and cost associated with the feature extraction process. This would enable an adequate comparison to be made with the software development as part of this project and provide as key input in aligning with the accuracy required for this project. (Flood 2004) discusses on ASPRS guidelines for vertical accuracy reporting of Lidar data. Effect of Lidar data density on DEM accuracy is detailed by (Liu *et al.* 2007). (Stoker *et al.* 2016) discuss evaluation of single photon and Geiger mode Lidar for 3D elevation program. Linear Lidar versus Geiger-mode Lidar - impact on data properties and data quality by (Ullrich and Pfennigbauer 2016) provided a background material should the findings from this project be expanded to include future state-wide capture using Geiger Mode Lidar methods. Modelling vertical error in Lidar derived DEM has been dealt by (Aguilar *et al.* 2010).

Methods to assess the performance of a Lidar algorithm were available in literature. These include:

- *Global and local context* – spatial coverage possible, larger the better (Sithole and Vosselman 2003a);
- *Computational efficiency* – Time taken to perform filtering (Sithole and Vosselman 2003a);
- *Data structure* – in (Sithole and Vosselman 2003a) single data structure represented by the profiles and line segments used for both segmentation and classifications with no fall back on other data structures nor other support data derived, assists to speed

up the algorithm; in (Sithole and Vosselman 2003b), some filters work on raw point clouds while some work on resampled image grid;

- *Verticality* – ability to handle surfaces lying vertically above each other (Sithole and Vosselman 2003a);
- *Adaptability* – for solving different detection tasks (Sithole and Vosselman 2003a);
- *Point density* – evaluation of filter performance based on Lidar point density (Vosselman 2000);
- *Lidar data noise* – accounting for Lidar data noise and final data precision (Vosselman 2000);
- *Type I vs. Type II Errors* – Errors in Commission or Omission (Sithole and Vosselman 2003b)
- *Performance in Steep Slopes* – Different performance criteria to flat terrain (Sithole and Vosselman 2003b)
- Working around special features (such as bridges) – (Sithole and Vosselman 2003b)
- Assessment of outliers – (Sithole and Vosselman 2003b)
- Performance on areas with vegetation on slopes – (Sithole and Vosselman 2003b)
- Effect of Lidar resolution – (Sithole and Vosselman 2003b)
- *Test neighbourhood* – Filters operate on a local neighbourhood; Algorithms can perform three kinds of comparison: *Point-to-point* (compare known point to classify unknown point); *Point to Points* (compare known point to classify unknown points); *Points-to-points* (compare known points to unknown points) (Sithole and Vosselman 2003b);
- *Measure of discontinuity* – “most algorithms classify based on some measure of discontinuity. Some of the measures of discontinuity used are, height difference, slope, shortest distance to TIN facets, and shortest distance to parameterised surfaces” (Sithole and Vosselman 2003b);
- *Filter concept* – “every filter makes an assumption about the structure of Bare Earth points in a local neighbourhood. This forms the concept of the filter”: *Slope-based* – slope or height difference between points measured and classified based on a threshold; *Block-Minimum* – Horizontal plane with specified buffer zone to classify points in or out of buffer; *Surface-based* – parametric surface with corresponding buffer zone to identify bare-earth points; *Clustering / Segmentation* – if points

cluster above its neighbourhood then it must belong to an object (Sithole and Vosselman 2003b)

- *Single vs. Iterative processing* – Recursive vs non-recursive with advantages in computational speed for single pass versus accuracy in multiple pass (Sithole and Vosselman 2003b)
- Replacement vs. Culling – (Sithole and Vosselman 2003b)
- Use of first pulse and reflectance data – (Sithole and Vosselman 2003b)

There are multiple papers that were explored to find methods of utilising a fusion of Lidar and Imagery. Table 2-2 shows some studies that have utilised an imagery and Lidar fusion feature extraction method.

Table 2-2: Studies with feature extraction using Image and Lidar fusion method

Developed by	Method
(Cheng and Weng 2017)	Urban road extraction from combined high-res sat image and ALS
(Du <i>et al.</i> 2016)	Building change detection using old aerial images and new Lidar data
(Gerke and Xiao 2014)	Fusion of ALS point clouds and images for supervised and unsupervised scene classification
(Hermosilla <i>et al.</i> 2011)	Evaluation of automatic building detection approaches combining high resolution images and Lidar data
(Hu, X. <i>et al.</i> 2004)	Automatic road extraction from dense urban area by integrated processing of high-res imagery and Lidar
(Kim and Medioni 2011)	Urban scene understanding from aerial and ground Lidar
(Meng, Currit, Wang, <i>et al.</i> 2010)	Object-oriented residential building land-use mapping using Lidar and aerial imagery
(Peng and Zhang 2016)	Building change detection combining Lidar and ortho image
(Rottensteiner <i>et al.</i> 2003)	Detecting buildings and roof segments by combining Lidar and multispectral images

(Schenk and Csathó 2002)	Fusion of Lidar and aerial imagery for a more complete surface description
(Sohn, Gunho and Dowman 2007)	Data fusion of high-res satellite imagery and Lidar for automatic building extraction
(Wang, H. and Glennie 2015)	Fusion of waveform Lidar and hyperspectral imagery for land cover classification
(Wang, L. and Neumann 2009)	Automatic registration of aerial images with un-textured aerial Lidar data
(Zhou and Zhou 2014)	Seamless fusion of Lidar and aerial imagery for building extraction
(Robinson <i>et al.</i> 2014)	Multi-scale smoothed, 90m digital elevation model from fused ASTER and SRTM data
(Huang <i>et al.</i> 2011)	Information fusion of aerial images and LIDAR data in urban areas: vector-stacking, re-classification and post-processing approaches

3 Study Area and Data Acquisition

3.1 Background

Lidar and Imagery of various resolutions were acquired at selected test sites. This section discusses the rationale behind the selection of those test sites and the data acquired.

Two sites with rural and urban characteristics were selected to ensure that the developed fence-line detection algorithm was able to detect fences for different kinds of built-up areas. It was assumed that the terrain and fence-lines in the two selected areas would serve as representative site for most other areas in Australia or New Zealand.

3.2 Terrain characteristics

Natural and man-made features such as vegetation, power-lines, open spaces, road structures and other man-made features are present in both the suburban and rural areas. The two selected sites have differing elevation changes in the terrain, and the fence-lines have their own characteristic differences. In both the areas, fences with different construction materials such as wooden, metal colour-bond and chain-wire fences are present. Fences are wholly or partially visible, and fences have hedges running alongside them, which has led to algorithms misidentifying the fence-lines.

The purpose of looking into variations in the study area is to train the future fence detection algorithms to work as efficiently in multiple environmental conditions. While designing data capture strategy for this project it was considered appropriate to design strategies that best suited the upgrade requirement by extracting fence-lines in both rural and urban areas.

3.3 Study area

Two representative rural and urban study areas were identified for the workflow development. Morayfield in the north of Brisbane was identified as a representative test area in a semi-urban environment, while Toowoomba in the west of Brisbane was selected as a representative area for a rural test site.

3.3.1 Semi-urban test area: Morayfield

This area is representative of a semi-urban environment with relatively smaller property sizes with distinct fence-lines. An outline of the test areas is shown in Figure 3-1 (a) and (b) with the location of the study area and an image taken over the study area.

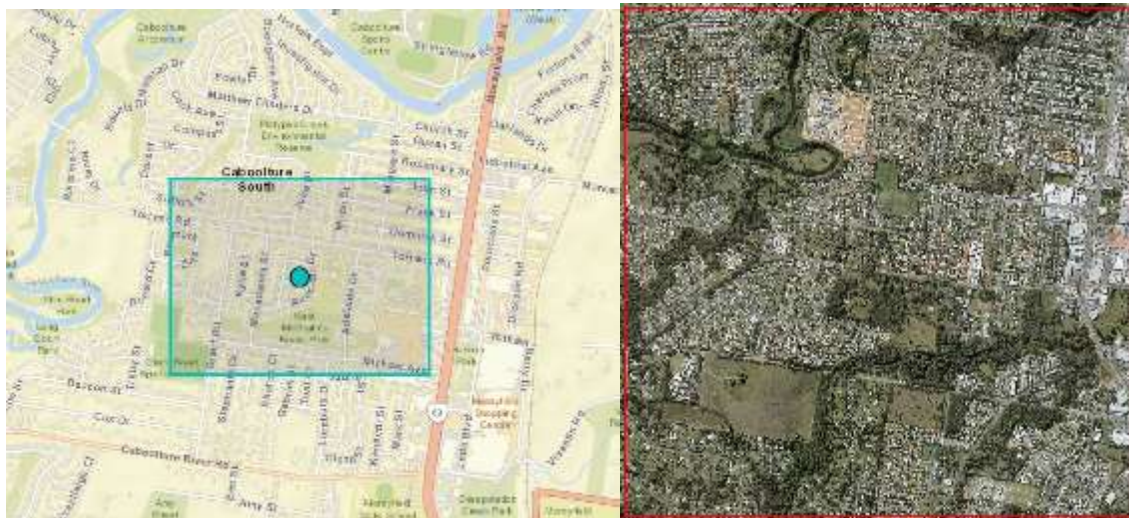


Figure 3-1: (a) Extent of the Project area in Morayfield, and (b) Aerial image of the Project area

3.3.2 Rural test area: Toowoomba

A second pilot area, representative of rural property is in the outskirts of Toowoomba, a city west of Brisbane. The area consists of relatively larger property sizes with different type of fences (usually posts connected by metal fences) compared to the Morayfield test site. An

outline of the test areas is shown in Figure 3-2 (a) and (b) shows the location of the study area and an image taken over the study area.



Figure 3-2: (a) Extent of the Project area, (b) Aerial image of the Project area

3.4 Existing Data

Lidar and Imagery data was available over the Toowoomba pilot area from the archive held by DNRME. Free test data was made available by RPS Australia of 24ppsm including stereo image for the initial development of the processing workflow while waiting for the data acquisition over Morayfield.

The project initially used the following data for workflow development, however since one of the objectives of the project was to evaluate feature extraction at various resolutions, it was decided to acquire Lidar and Imagery over Morayfield (see Section 3.5) to ensure that data captured at the same flight over the same area was utilised for evaluation:

Morayfield initial data:

Aerial Lidar: There were 8 tiles of LIDAR point cloud provided with a point density per tile that varies between 4-8ppsm. Fusing of point cloud data from 8-Lidar tiles thus yielded a combined point density of approximately 24ppsm.

Aerial Photo: There are 72 RGB aerial images with a GSD of 8cm. The photos have adequate overlap for bundle block adjustment and have the required interior and exterior orientation parameters.

Toowoomba:

Aerial Lidar: The Lidar point cloud consists of a single tile covering approximately 30 sq.km (5km X 6km area) where the average point density is approximately 2-3ppsm.

Aerial Photo: An ortho-photo at 10cm GSD available for the area was clipped to match the extent of the LIDAR tile.

Adelaide, South Australia:

Aerial Lidar: The Lidar point cloud consists of multiple square tiles over the CBD of Adelaide with an approximate point cloud density of 20ppsm.

Aerial Photo: An ortho-image probably captured simultaneously with the Lidar data was provided to the project along-with the digital cadastre and control point vector shapefiles.

Geiger Mode Lidar, USA:

Aerial Lidar: A Geiger Mode Lidar point cloud over USA was made available for a small area with an approximate point cloud density of 32ppsm.

Aerial Photo: A small imagery file of low resolution that marked the area of interest in graphics was provided, however as this was of a very low resolution, this was not used.

3.5 Data acquisition

Additional high-density airborne Lidar data using Trimble AX60i sensor (Appendix 10.3) was captured over the Morayfield area. The combined point density for the multiple flight lines resulted in most areas with a density of 50 to 100 points per square metre.

Multiple overlapping flight-lines were flown to achieve a higher density combined point cloud. Aerial imagery was simultaneously captured using AICP65 Pro camera with 6cm GSD that resulted in overlapping stereo imagery used for ortho-rectification.

(Figure 3-3 Left) shows the photo centres of imagery data capture and (Figure 3-3 Right) shows the flight lines. There are 34 flight lines in total, 17 each in the north-south and east-west direction. A total of 68 Lidar scenes were captured while the total number of imagery captured was 1156.

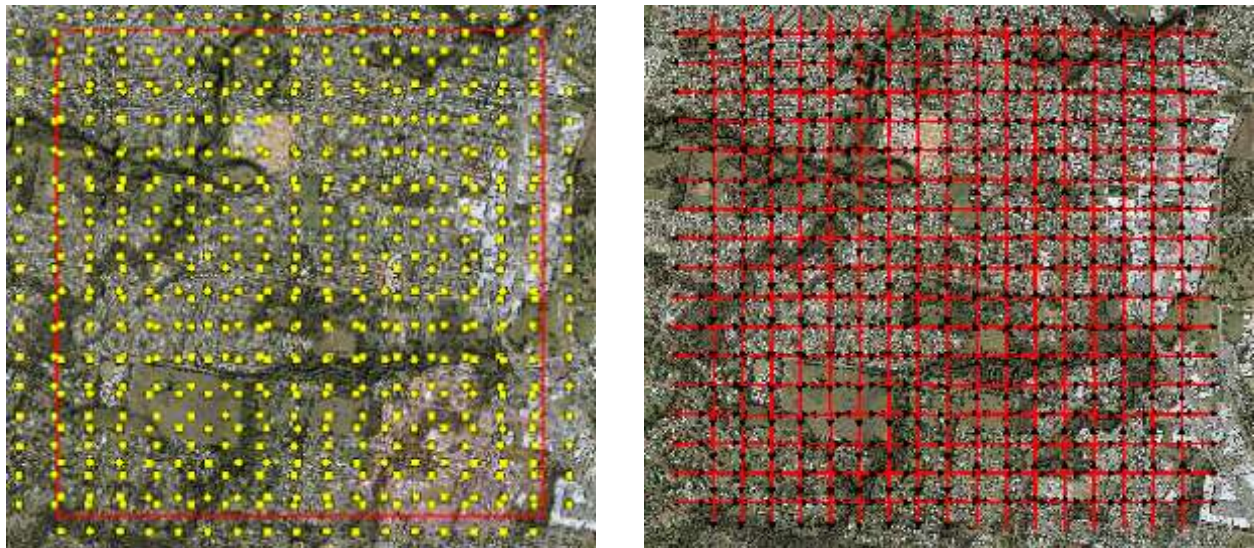


Figure 3-3: (Left) Photo centre over Morayfield for aerial imagery data capture, (Right) Flight lines for Lidar and Imagery data capture

3.6 GPS Field survey

After the Lidar and Imagery data capture, GPS field survey (Figure 3-4) was done to verify the accuracy of the Lidar data and for ortho-rectification of stereo imagery. Five locations that could be identified in the imagery were selected. These selected locations were flat planes of around 1m² to enable Lidar data validation as per ASPRS Guidelines for vertical accuracy reporting for Lidar Data (Flood 2004). In addition three permanent marks were selected to tie it to the national datum and the CORS network of Caboolture used for initial data processing. The final data processing was done using AUSPOS solution (Table 3-1).



Figure 3-4: GPS Field Survey locations

Table 3-1: Extract from AUSPOS Solution for GPS Field Survey Calculations

Station	East (m)	North (m)	Zone	Ellipsoidal Height (m)	Derived AHD (m)
0643	494144.660	7000577.050	56	53.779	10.861
0753	494764.765	7001017.771	56	59.348	16.431
5154	494900.611	7000646.300	56	58.029	15.122
TGT1	494903.245	7000645.653	56	58.100	15.193
TGT2	494228.762	7001027.850	56	51.817	8.891
TGT3	494604.607	7001323.800	56	51.500	8.575
TGT4	494766.851	7000993.469	56	59.753	16.836
TGT5	494142.412	7000575.016	56	53.772	10.854

MGA Grid, GRS80 Ellipsoid, GDA94



Reference Stations used for the AUSPOS solution

Date	User Stations	Reference Stations	Orbit Type
2018/05/31 01:18:30	0643 0753 5154 TGT1 TGT2 TGT3 TGT4 TGT5	ALIC BDST CBLT CEDU CLEV GATT HOB2 IPSR MOBS ROBI SYDN TID1 TOOC TOOW TOW2	IGS final

3.7 Image Ortho-rectification

Image ortho-rectification was done for captured data in Morayfield in a software called Icaros One Button using camera parameters shown in (Appendix 10.1) and photo centre coordinates from on-board navigation systems.

The Imagery was captured at 6cm GSD and had multiple overlaps from both north-south and east-west flight directions (Figure 3-5 Top). The ortho-rectified image using the supplied photo-centre coordinates showed that there was a systematic shift of 1.2m (Figure 3-5 Middle) in all the images with respect to data captured by Lidar systems. As both Lidar and Imagery had been captured at the same time from the same plane using the same reference CORS station, the shift could be attributed to the accuracy attained by the GNSS system for the camera.

The images were ortho-rectified again using coordinates obtained from the GPS field survey and DEM from Lidar, and the result of the ortho-rectification showed good match with lines obtained from Lidar (Figure 3-5 Bottom).





Figure 3-5: Image capture and processing (Top) Aerial photo of 6cm GSD; (Middle) Systematic shift noticed during ortho-rectification; (Bottom) Ortho-rectified imagery using GPS ground control

Discussion: The initial ortho-rectification demonstrated that there is a potential for error through systematic shift during ortho-rectification which can lead to incorrect positioning of cadastral data. It is not certain whether it is quite common for ortho-rectified imagery supplied to the department to have these systematic errors or whether they have been ortho-rectified using ground control and errors minimised.

Therefore, it is recommended that imagery should be ortho-rectified using GPS field survey coordinates and high-resolution DEM preferably obtained from Lidar to improve the results of feature extraction or data validation as demonstrated by the improved result of line-fit based on the new ortho-rectification.

4 Methodology

4.1 Background

This section addresses objective one of this research. High density Lidar data complemented by high-resolution imagery was used in the development of an independent as well as an integrated workflow for extraction of fence-lines to be used for DCDB block-adjustment. The data for the project were sourced from multiple sources including conducting a flight for data acquisition for high-resolution latest data.

The project draws on existing expertise and past research outputs from the CRCSI's "Feature Extraction" Program, other existing research, and ESRI Australia and Harris Geospatial's expertise. This existing body of knowledge was adapted to apply and extend these capabilities to optimally address the needs of the project.

The specific requirements of this project was to extract fence-lines which are narrow linear features at a given elevation range and have a range of characteristics. Empirical assessment of the different data sources in various combinations was undertaken based on both cadastral data and ground truth information.

Various candidate 2D and 3D feature extraction tools were selected and evaluated before finalising an independent workflow for Lidar data plus an integrated workflow for imagery and Lidar data which extracted fence-lines to be used form block adjustment of digital cadastral data.

The evaluation of the data, the workflow, and the feasibility of utilising the results for cadastral upgrade included the determination of geo-positional biases between the digital cadastre and land parcel boundary segments extracted via the feature extraction approach for fence-lines.

4.2 Research Steps

4.2.1 Project set-up, Literature Review and Planning

A comprehensive literature review of research into feature extraction from Lidar and Imagery was undertaken. It also included existing work in upgrading the cadastre via imagery and Lidar, from aerial & space-borne platforms.

This review of research revealed that while there are numerous existing feature extraction methodologies and applications that have been used to delineate linear features, yet there are no research that have focussed on extracting the fence-lines with a view to move the graphical representation of the digital cadastral boundaries with geo-position inaccuracies to the ground positions accepted by landowners.

Further, the existing research revealed that while there have been capabilities to extract a power-line, there have been no research to explore the feasibility of extracting a narrow feature such as a fence that may be built using various construction methods, have various heights above the ground, often have hedges running along it, and are often obscured by existing trees and man-made structures.

4.2.2 Accuracy requirements for Queensland cadastre

An investigation into the existing accuracy of Queensland cadastre, the accuracy requirements associated with cadastral upgrading, and that achievable through the adopted methodology was undertaken.

The starting point for exploring the workflow was through high-resolution Lidar data. After multiple algorithms and workflows and been explored, the adopted workflow was tested for different resolutions of Lidar data and an integrated Lidar plus Imagery workflow. High-resolution Lidar data and Imagery were acquired of appropriate resolution and geo-positional accuracy. High-resolution Lidar data has demonstrated a capacity to identify linear features such as power-lines. Thus, it was expected that a similar resolution Lidar would assist in identifying fence-lines, and high-resolution imagery can be used as a complement to produce

an integrated boundary detection tool. However, it is noted that this detected boundary lines may or may not coincide with a property boundary.

4.2.3 Study area identification

Identification of the pilot project area and associated data sources, were done carefully noting the requirement to include both urban and non-urban environments and the desirability of having access to multiple sources of data.

Also, at least one of the areas needed to be in a location where it would be feasible to acquire suitable imagery and Lidar data while maintaining the selected terrain characteristics, noting that resolution requirements and accuracy tolerances in rural areas need not be as stringent as those in cities and towns.

4.2.4 Manual feature extraction and accuracy analysis

An initial manual feature extraction and upgrading exercise was performed, the aim of which were:

- i. To validate the overall upgrading workflow envisaged for the semi- and fully automatic method being developed and tested;
- ii. To assess the metric performance of the various data sources, and verify that the accuracy of the feature extraction could be comparable to a relatively spatially accurate cadastre;
- iii. To assess the feasibility of block adjusting a distorted cadastre into its expected spatial location using the extracted fence-lines and derive metrics of evaluation of accuracy;
- iv. To provide the benchmark or control data against which the semi- and fully automatic methods were evaluated.

4.2.5 Evaluation of alternative methods

Several different methods of feature extraction from Lidar, Imagery and Integrated methods were explored. The results were evaluated and some of the methods were discarded while parts of some of the methods were used for the final processing workflow that was adopted.

4.2.6 Processing pipeline and software development plan

This phase involved identification of the series of manual steps for the development of software to support the automated extraction of cadastre-relevant features from the imagery and Lidar data, with appropriate analytical functions to quantify geo-positioning discrepancies with respect to the existing digital cadastral data. Several methods were explored and finally a workflow for Lidar and another for integrated Lidar plus Imagery was finalised with an overall workflow shown in Figure 4-1, and detailed processes shown in Figure 4-14 and Figure 4-15.

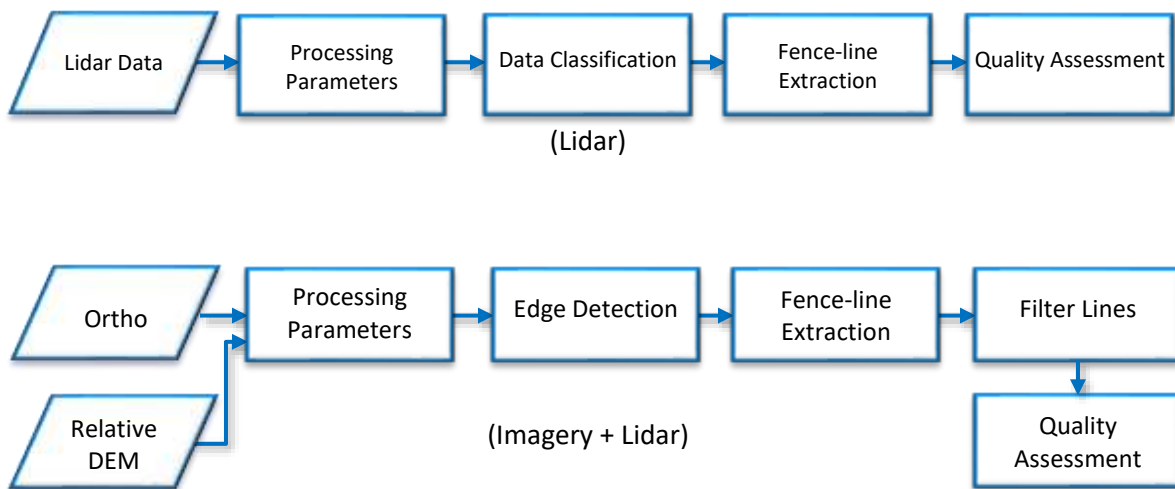


Figure 4-1: Processing overview for (Top) Lidar data and (Bottom) Integrated Imagery and Lidar data

4.2.7 Automated feature extraction and data analysis

A workflow/processing pipeline to support the application of the adopted (semi/fully) automated feature extraction tool for the extraction of fence-lines was developed and packaged in a graphical user interface (GUI).

The new software developed to support the automated feature extraction/upgrading stage included an independent process for feature extraction from Lidar and an integrated feature extraction method from Lidar and Imagery.

The output of the feature extraction process primarily comprise of linear features forming initially non-concatenated boundary segments represented by fences which can be used as is or cleaned to extend to the intersections from which cadastral lines or polygons can be block-adjusted. The extracted line features are then be compared to the current cadastral database to ascertain the spatial accuracy of the extraction as well as the cadastre and to quantify corrections, mostly geo-positional biases, which need to be applied to the cadastral data.

4.2.8 Evaluation of feature extraction

An experimental evaluation of the semi- and fully automatic feature extraction and cadastral upgrading methodology was undertaken for the developed workflow over a range of different sites, from urban areas through to rural properties.

The purpose here is to fully assess the developed workflow in terms of its practicability, accuracy, completeness, and general reliability as a means of automated upgrading of the cadastre to the required levels of accuracy.

4.2.9 Reporting and recommendations:

This document is prepared to report on the outcomes of the research. This addresses:

- i. Achievable accuracy from available data sources used in isolation and in combination;
- ii. Recommended software tools and capabilities;
- iii. Technical challenges and limitations of the approaches used.

The report also makes recommendations on future work and options for operational implementation.

4.3 Methods of Feature Extraction Explored

4.3.1 Overview of Methods Explored

Several methods were explored for ways to extract fence-lines, either from Lidar, or Imagery or combined. The processing methods on their own had some things that worked and some things that did not, so the learnings from what worked and what did not was used to develop a processing workflow that was further used to create a GUI for Lidar processing and an Integrated Lidar and Image processing workflow. The various processes explored are listed in Figure 4-2 below and further details about the processes are discussed in Sections 4.3.2 to 4.3.8.

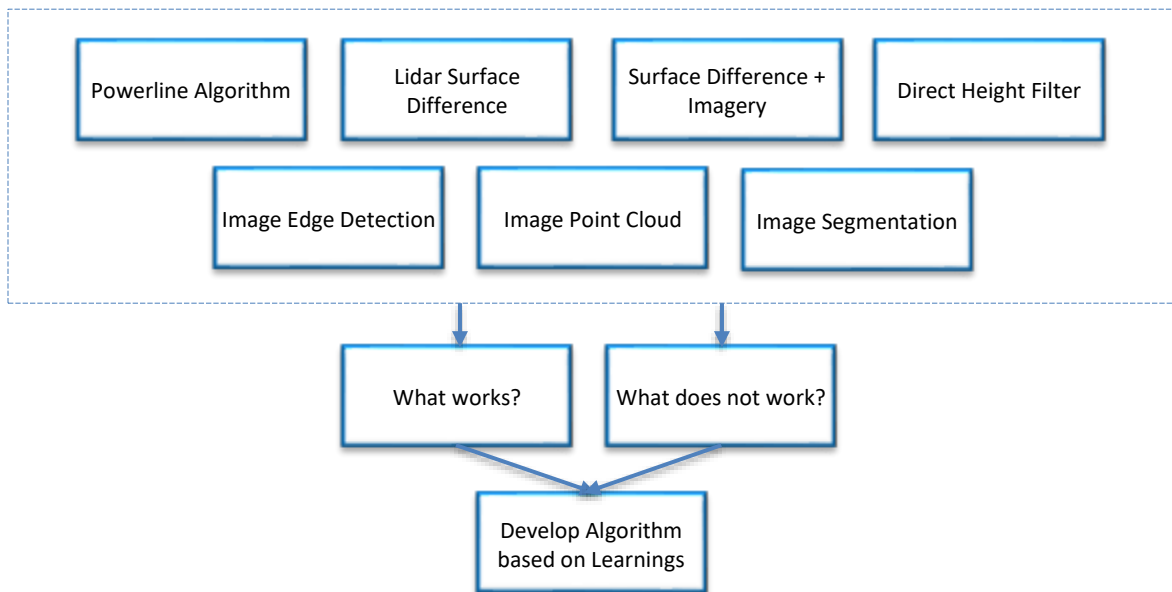


Figure 4-2: Various processes explored for fence-line extraction

4.3.2 Method using Power-line Extraction Parameters

Power-lines vectors can be identified to be an entity closest to fence-lines in terms of narrow elongated features with specific height attributes. The key difference is that power lines are much taller than a typical fence line and can be extracted as a continuous linear feature with clear return numbers in Lidar data capture.

There are usually very little differences in construction material for power-lines compared to fences, hardly any confusing vegetation running alongside it as in hedges running along fences, and geometrically, fence-line vectors are relatively shorter in length and often connected at right angle to each other or incomplete at the front of the houses.



Figure 4-3: Workflow for Powerline Extraction Algorithm

The first step in the extraction process was to extract all the typical features in the scene except for powerlines and leave rest of the points as unclassified that could potentially represent fence lines (Figure 4-3). The next step was to filter out any unclassified point cloud above or below 2.0m to 0.5m respectively as this is the likely range of height above ground for fences. A powerline extraction algorithm was run in ENVI and the results exported to vector files for quality assessment.

The algorithm does a decent job of identifying fence-lines in the raster version, but it also has a large number of omissions in its fence-lines detection and missing lines in the vector format (Figure 4-4 and Figure 4-5). It also identifies a number of other features at the same elevation range (false positives), and since the powerline algorithm is a black-box in the ENVI environment, it is limited to be used in an ENVI environment. Therefore, parts of the process

was used for the final workflow which was developed in an open source IDL as well as an ENVI IDL version.



Figure 4-4: Points in white represent fence but other unclassified points are classes as fence as well



Figure 4-5: Fences detected using powerline algorithm showing false positives and omissions

4.3.3 Lidar Surface Difference Method

Digital Surface Model (DSM) represents the elevation of features on the surface along with elevation of the ground, while Digital Elevation Model (DEM) represents the elevation of the ground in the scene.

The fence-lines extracted from various methods have false positives along the ground such as kerb lines and other low lying features. One option to get rid of these false positives is to use remove the ground level information from the elevation models. This involves subtracting DEM from DSM to create a model that hold just the surface elevation information where each pixel represents the elevation of the underlying pixel (Figure 4-6). This can then be filtered according to the range of fence heights and exported.

The problem with such an approach was due to software limitations where DEM and DSM were extracted at different pixel resolutions resulting in loss of information due to varying resolution. Additional steps were also required to bring them to the same resolution and there were difficulties in additional filtering for features other than fences. It was also difficult to cluster the point cloud to a single fence-line vector and to ensure that fences were selected and no other ancillary objects such as hedges. This process however had its merits with the surface difference model and this idea was implemented in the final workflow.



Figure 4-6: Surface difference from DSM-DEM derived from Lidar

4.3.4 Lidar Surface Difference on Imagery Output

Edges detected from image based feature extraction could be used in combination with surface difference model to improve the results of the extraction by removing noise. There is a marked improvement in the result as many false positives on the ground and within the building envelope are now eliminated as relatively accurate object heights can be used to eliminate ground features such as roadside kerbs (Figure 4-7). Parts of this method was used in the integrated image and Lidar based method developed as a GUI.

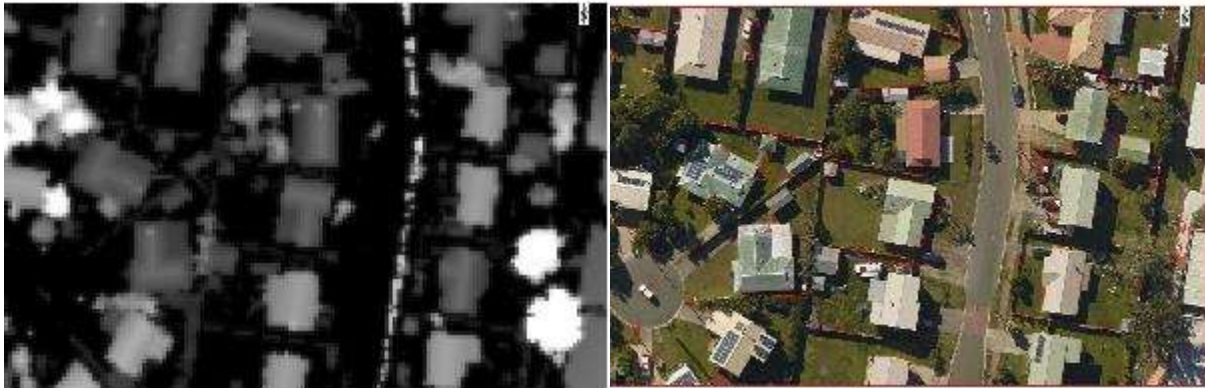


Figure 4-7: (Left) Lidar surface difference model, and (Right) Extracted fence-lines on imagery

4.3.5 Lastools Direct Height Filter Method

Appendix 10.5 describes the approach used to filter out fence-line raster (Figure 4-8) using methods described in individual tools in Lastools documentation. These processes provide good results in raster and is necessary for developing part of the input raster for use in the Open Source IDL GUI developed for the project.

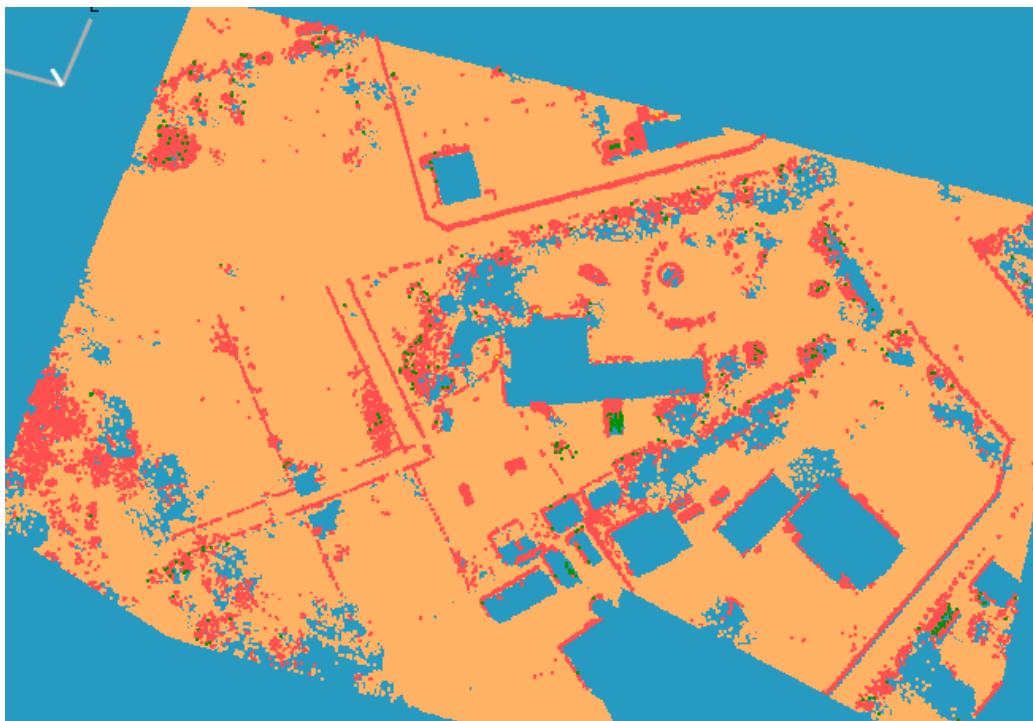


Figure 4-8: Fence-line raster filtered using Lastools

4.3.6 Image Edge-Detection Method

Various options for edge detection from imagery are available. Figure 4-9 shows the result of edge-detection operation on a single band of the RGB image. A large number of features, trees and most of the linear features are extracted from the image. There are some false positives that can be removed by superimposing either the building raster or building footprints (Figure 4-9 Bottom). One disadvantage of this method is inconsistency of results based on image characteristics, capture of shadows, and displacement of fence-lines due to look angle.

Parts of this method was used to develop the integrated Imagery and Lidar fence-line extraction method in the GUI.



Figure 4-9: (Top) Edge detection shows a large number of edges; and (Bottom) Buildings overlaid on the edges

4.3.7 Image-derived Point-Cloud Filter Method

Dense point clouds can be generated using image matching methods using photogrammetric methods to take advantage of forward and side overlap in images and known acquisition geometry (Figure 4-10).

One important reason for investigating this aspect in this research is it explore whether it will help to substitute and/or supplement aerial Lidar derived point cloud in areas where no such capture is available. An initial assessment was made to judge the suitability of utilising photogrammetric point cloud derived from aerial images.



Figure 4-10: Dense point cloud generated from stereo-pair images by image matching

Figure 4-11 shows the results of feature extraction of fence-lines using point clouds based on aerial imagery. Initial assessment of the resulting fence-line shows many false positives and far more omissions. For image-based point clouds, viewing geometry can cause difficulty in image matching. Furthermore, shadows often associated with the fence-lines and a lack of contrast with the immediate background cause issues with this extraction method. It was thus concluded to not explore this any further, and that while using imagery, image-based feature extraction may be better suited than derived point-clouds for feature extraction.



Figure 4-11: Feature extraction from aerial image derived point cloud

4.3.8 Image Segmentation Method

Image segmentation was explored in ENVI software using an ortho-rectified imagery. Object based Image analysis (OBIA) provided benefits such as additional segmentation parameters for image analysis such as texture, spectral and spatial attributes.

The spectral brightness of the fence-lines, the spatial properties such as area, elongation and length of the features; and the texture of the image were used to extract the fence-lines (Figure 4-12). The results varied between different areas and different parameters had to be determined for different areas based on several iterations of what worked for that image and area. The resulting fence-line extraction had false positives and omissions (Figure 4-13). Therefore, this method was not explored any further.

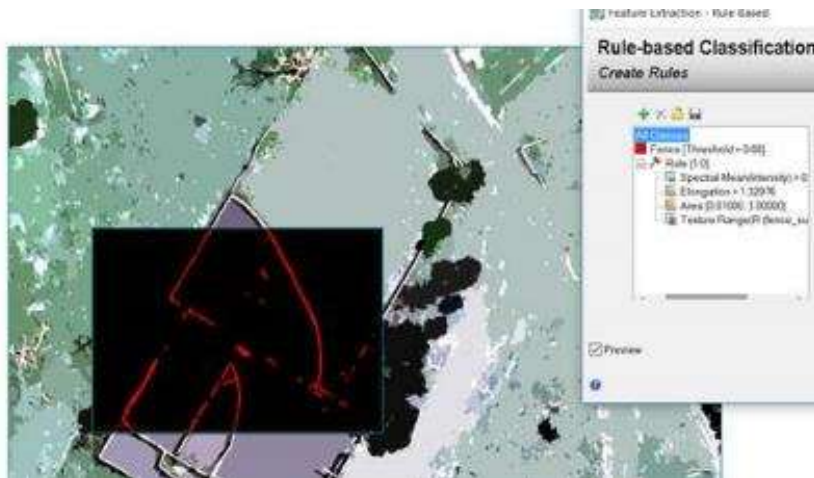


Figure 4-12: Image segmentation rule creation window with moving overview window



Figure 4-13: Fence-line extracted using image segmentation with false positives and omissions

4.4 Considerations for Fences and Corresponding Algorithms

From exploration of the various methods for feature extraction, it was determined to use parts of the methods that worked. It was thus necessary to define the fence characteristics that would be necessary to be considered for further development of an algorithm. This section discusses the characteristics of fences and how these characteristics are considered in the algorithm (Table 4-1).

Table 4-1: Fence-line characteristics and algorithm considerations

Fence-line characteristics	Algorithm consideration
<ul style="list-style-type: none"> a. Fences are long and relatively thin features b. Fences may have gaps in data due to inherent gaps (gates etc.), visibility gaps (fence blocked by vegetation, buildings or sheds) 	<ul style="list-style-type: none"> 1. Use an elongation ratio (length/width) threshold 2. Select minimum length tolerance to eliminate segmented and spurious lines 3. Use a maximum gap tolerance along identified fence-lines for line fitting 3. Eliminate larger areas in the processing that are not a result of (large length value X small width value)
<ul style="list-style-type: none"> a. Fences have hedges growing next to them; b. Fences can be made of hedges 	<ul style="list-style-type: none"> 1. Use the RANSAC algorithm to select the most probable line with a maximum cluster distance threshold 2. Utilise vegetation removal kernel radius threshold in 2D beyond which points are not considered 3. Iterate to test if points eliminated previously can be included between the segments
<ul style="list-style-type: none"> a. Fences generally have elevation ranging between 0.5m to 2.0m b. Fence-lines may be confused with other linear lines such as buildings, powerlines, road c. Fences may have high vegetation covering 	<ul style="list-style-type: none"> 1. Eliminate lines formed on bare earth, buildings or high vegetation; 2. Use a filter that selects point cloud between 0.5m-2.0m for the analysis 3. Identify and remove buildings, powerlines 4. Identify and remove trees

<p>a. Fences are usually in cardinal directions but can be any direction in between;</p> <p>b. Fence joins are close to 90 degrees</p>	<ol style="list-style-type: none"> 1. Use directed convolution filters to identify edges in cardinal and diagonal directions (imagery); 2. Calculate curvature to eliminate non-straight lines 3. Eliminate joins with an angle greater than a specified threshold
<p>There are different types of fences (e.g. continuous paling fences, hedges, post and wire fences etc.)</p>	<ol style="list-style-type: none"> 1. Different fences return different concentration of Lidar point clouds, so use different settings for line gaps etc. in the .json files
<p>a. In flat surfaces, fences usually have the same height throughout a single line;</p> <p>b. Cars etc. at a similar height range have a plane surface</p>	<ol style="list-style-type: none"> 1. Use a z-component of plane fit over normal vector, i.e. plane fit over original point clouds 2. Use a tolerance threshold to eliminate points outside a given z-plane 3. For fences on sloping ground where the z-plane changes rapidly, not use this component

The choice of filter/kernel and its parameters used to detect the fence-line features can have varying degrees of influence on the accuracy of the extraction. After exploring the relative merits of the various aspects, the following factors were selected that formed the core of the algorithm development:

- Maximum height above terrain to look for fence points
- Minimum height above terrain to look for fence points
- Minimum number of points in a cluster to be considered to be part of a fence
- Distance between points to fit linear features
- Maximum gap along fence to fit lines
- Minimum length of fence-line segments
- Kernel radius to remove vegetation near fence-lines and its threshold
- Exclusion of points as fence-lines based on previous classification
- Parameter for adding back a point if a line is subsequently detected
- Distance threshold for points for fitting a plane (to extract fence-lines)

4.5 Workflow developed for Fence-line detection using Lidar

A linear workflow was developed for the fence-line extraction. The philosophy behind the development of this workflow was to make it as semi/fully – automated as possible and also to provide users various options to choose algorithms and parameters that suited the available Lidar point-cloud density.

Three parameter files were developed based on the point cloud density of the available data: Low/sparse density; Medium density; and High density, that broadly reflects real-life data capture scenarios. The workflow also caters for advanced users to modify fence-line extraction parameters as an iterative process to improve the overall extraction results.

An important consideration in the design of the workflow (Figure 4-14) is to cater for varying quality and geographic coverage of the data that is likely to be used in a jurisdiction. Thus, variability in point densities was addressed by developing algorithms that is parameterised using a model that caters for three densities described in Table 4-2. The three different parameter files are dependent on the density of Lidar points per square metre (ppsm) and various other considerations (More details in Appendix 10.6).

Table 4-2: Lidar Density vs. parameters to be used

Lidar Density	PPSM Range	Terrain type	JSON File to use
Low	> 4 ppsm	Mostly Rural	ParamSparseData.json
Medium	4 – 20 ppsm	Rural to Urban	ParamMediumDensityData.json
High	< 20 ppsm	Urban	ParamHighDensityData.json

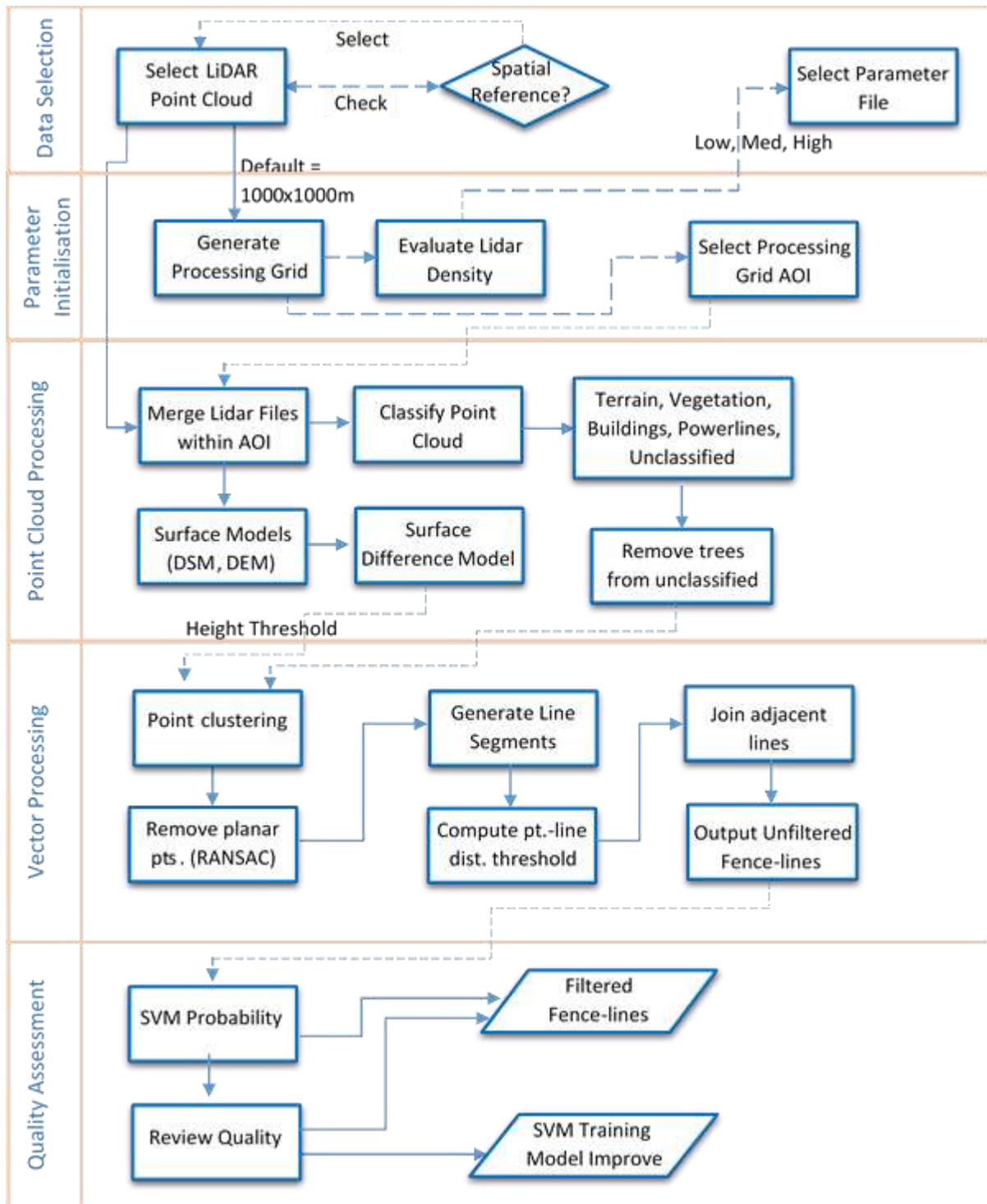


Figure 4-14: Workflow for Fence-line extraction using Lidar data

As the spatial extent of Lidar point-cloud coverage could be large, and to make allowances for various computational capacities, the algorithm provides an option to process these files by splitting them as rectangular processing grids defined by users to cater for the size of the area being used for fence extraction.

The workflow is also designed to refine the result with a SVM based self-learning model that allows users to guide the extraction process by training the model with user identified correct versus incorrect feature extractions to refine the final output.

The key steps in the workflow are described as follows:

- a. Select the input LAS file (or a collection of Las files), assess point-density which in turn would allow in the selection of the parameters of extraction;
- b. Split the LAS file into desired rectangular grid sizes and select a processing AOI
- c. Process the Lidar and extract fence-lines
- d. Review and refine the vectors and the training model and output the fence-line vectors

4.6 Workflow for Fence-line detection using Integrated Imagery and Lidar

The approach taken consists of multiple steps in order to derive line segments from imagery (Figure 4-15).

Edge Detection: Using the Canny edge detection algorithm, edges can be extracted efficiently from imagery. The image was processed in tiles of 256x256 pixels at a time. The Canny algorithm returns many more edges than are desired to be included. Most notably, vegetation and textured roofs returns high concentrations of edges.

Clustering: Clustering of adjacent (contiguous) pixels in the Canny edge detection output is performed next to identify and further process groups of pixels to see if they should be included or excluded. Each contiguous group of pixels is then passed in to the next step in the algorithm.

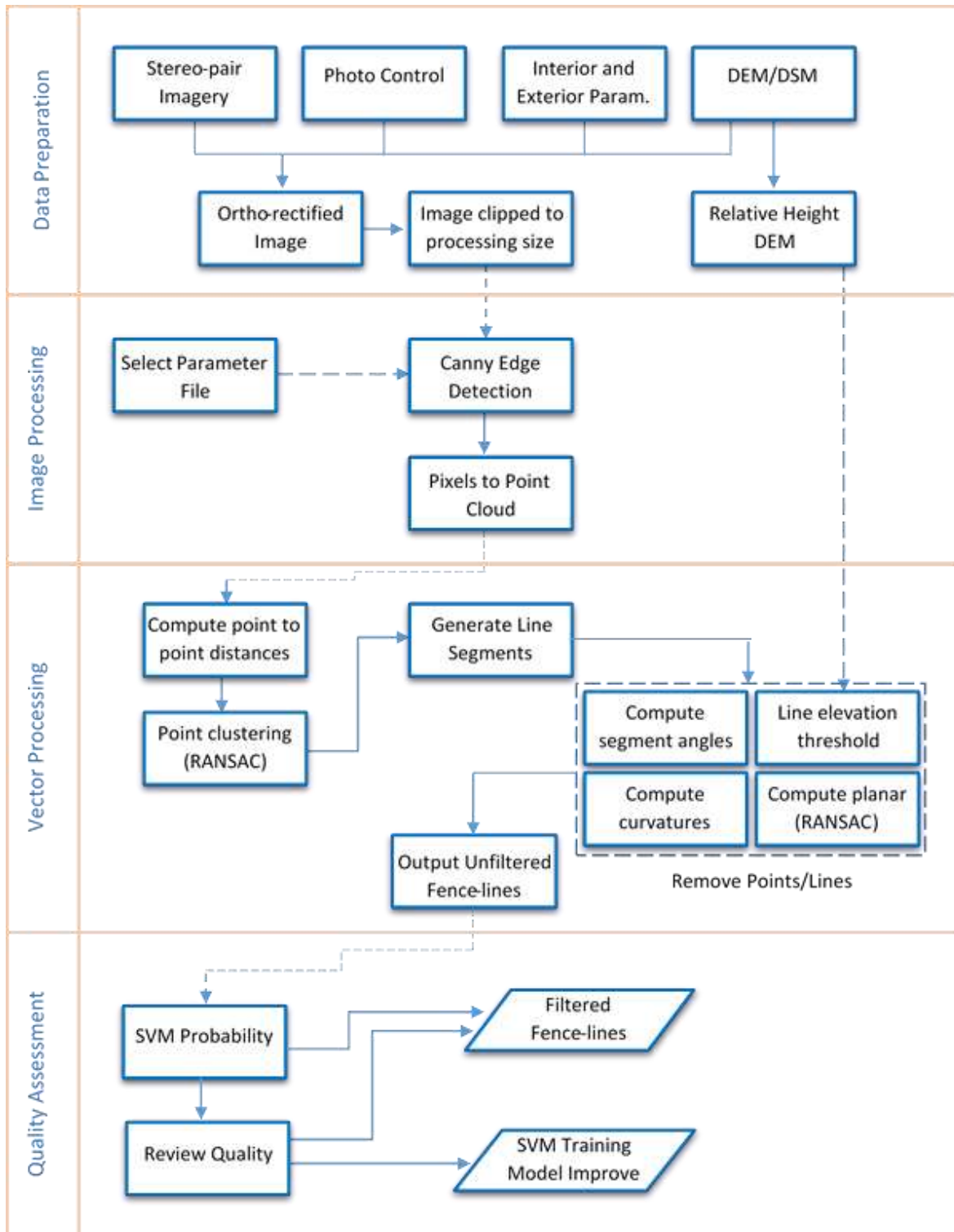


Figure 4-15: Workflow for Fence-line extraction using Ortho-imagery and Lidar Relative Elevation

Curve calculation: The amount of curvature is then calculated for each of the groups of inlier pixel locations computed in the previous step. The idea is to remove pixel groups that are too curved (i.e. vegetation, textured roof tops, etc.). The way the curvature is computed is to use the two nearest neighbours of each point (x, y) , then compute the angle between the two vectors towards the two nearest points. For a straight line, this angle will be close to 180 degrees, but for a curved section, this will be a smaller possibly closer to 90 degrees or even smaller. The amount of curvature is then calculated as the average angle for all the points in the given group. A threshold of 137.5 degrees was chosen here to only report pixel groups with a curvature angle larger than this value. The remaining pixel groups are shown in blue to the right in the figures below.

Modified RANSAC line search algorithm: The standard RANSAC line fitting algorithm takes a collection of points and chooses random pairs of points to create possible line fits, then using a distance threshold to define inliers vs. outliers, it searches for the line that gives the highest number of inliers (meaning points within the distance threshold). In the modified version used here the pixel locations are translated in a contiguous group, from the previous step, into a set of coordinates (x, y) points. Now, instead of searching randomly, every point with all points closer than a given radius are grouped to use as the search space for line definitions. Inliers and outliers are computed. A minimum number of inliers is needed to proceed with the line segment.

Vectorization: The inlier points (x, y) derived from the pixel locations and filtered in the previous step, are then converted into a line segment using a linear least squares fit of the remaining points.

4.7 Summary

This section outlined a research methodology for the project. Several methods of feature extraction were explored and based on that, a final workflow was developed for the development of a GUI using Lidar and integrated Lidar and Imagery inputs.

5 Data Processing and Evaluation of DCDB Upgrade

5.1 Background

This chapter addresses objective two of this research and it evaluates the contribution of the various data processing options explored in this research and the workflow developed as a result for the upgrade of cadastral data.

The discussions in this chapter can be described in the following distinct steps:

- Details the algorithm development, Lidar and imagery GUI development and SVM implementation;
- Assesses the results of the data processing for various geographic scenarios;
- Evaluates the feasibility of DCDB upgrade using the extracted fence-lines; and finally
- Assesses the results of various Lidar resolutions for fence-line extraction.

5.2 Fence-line Extraction Algorithm Development

5.2.1 Algorithm Processing Stages – Lidar

An algorithm was developed to extract fence-lines from Lidar data based on the factors that were identified. The intention was to add (or discard) additional factors based on the results as it was difficult to be precise on the actual vs estimated influence of each parameter through the extraction process. The final output was based on a probabilistic value for a particular line segment being a fence or otherwise. The algorithm includes multiple steps described below in Steps 1 to 10 and Table 5-1, while details on how to run the GUI is in Appendix 10.7.

- Step 1 – Combine/Subset LAS files into a Lidar project. Split Lidar dataset by area based on a set of Lidar files in .LAS format in combination with a shapefile containing polygons defining the areas of interest.

- Step 2 – Point cloud classification and DEM extraction. Create surface model and topographic shaded raster image. This step also involves classifying terrain (ground) points, vegetation (trees), and buildings, as well as generating a bare earth DEM raster.
- Step 3 – Create DEM terrain raster representation and a relative height raster output. This creates a height raster relative to the bare earth DEM at a higher resolution than the bare earth DEM.
- Step 4 – Hillshade output. This is creating a hillshade raster that is useful for visualization purposes only. This raster representation is not used for the fence extraction algorithm.
- Step 5 – Process fence specific algorithm in tiles to allow arbitrary project size. Points are processed in small tiles to facilitate locating of fence-line segments.
 - Step 5a – Filter points based on height range relative to DEM – apply height thresholds relative to ground.
 - Step 5b – Filter points based on classification (unclassified or trees). Apply class filter to exclude terrain points and buildings, include vegetation and unclassified points.
 - Step 5c – Perform distance based clustering to group points that belong to the same structure.
 - Step 5d – Remove tree points based on circular coverage criteria - for each cluster, remove circular objects such as trees.
 - Step 5e – RANSAC line finding and fitting algorithm used to extract line segments from points. For each cluster, perform a line fit (in X/Y, ignoring Z) and remove points that are too far from the line.
 - Step 5f – Remove clear non-fence points based on cluster dimensions (remove small clusters). For each line, check length and number of points against minimum thresholds.

- Step 6 – After looping over all tiles, combine line segments that have the same direction and are Adjacent. Join line segments based on proximity and direction avoid gaps from tiled processing in previous steps.
- Step 7 – Output all line segments along with statistical metrics for the points in each line segment. Filter short line segments based on minimum length parameter.
- Step 8 – Compute probability for whether each line segment is a fence or not a fence. This step uses an SVM (support vector machine) model to calculate the probability that a line segment is a fence.
- Step 9 – Output all line segments along with the probabilities.
- Step 10 – Output filtered line segments based on a probability threshold or manual edits.

Table 5-1: Algorithm parameters specific to fence extraction

Parameter	Description
Maximum_Height	Maximum height above terrain to look for fence points.
Minimum_Height	Minimum height above terrain to look for fence points.
Minimum_Num_Points	Minimum number of points in a cluster to be considered part of a fence
Perpendicular_Tolerance	Line fitting distance threshold.
Maximum_Gap	Maximum gap along fence during line fitting.
Minimum_Length	Minimum length of a fence-line segment.
Veg_Removal_Kernel_Radius	Kernel radius for filtering high coverage clusters of points (meters) before line fitting.
Veg_Removal_Grid	Number of grid cells in each dimension (2D)
Veg_Removal_Maximum_Coverage	Maximum coverage percentage, above which points are removed from consideration.
Expand_Point_Search	Boolean parameter to allow adding back in previously removed points after line fitting (default: false). If set to true, partial fence-line detections tends to become more complete

	at the expense of adding false positives.
Plane_Threshold	Distance threshold used for plane fitting of points (plane fit is used as statistical metric).
Plane_HReject	Boolean parameter indicating that points found to be on a near horizontal plane should be removed from consideration prior to line fitting.
Cluster_Distance	Distance in meters used to group points belonging to the same object.
Classes	A list of classes (from a previous classification) used when looking for fence points (terrain and buildings should be excluded).

Discussion: Considering the output of the fence-line extraction, it was considered more important to maximise the true positive detection rate than removing all false positive detections.

While it was found that including all the vegetation in addition to the unclassified points resulted in a much better detection of fence-lines in suburban areas, but it also resulted in many false positive detections in wooded areas. Thus, a consideration for the algorithm was to remove points that are part of a tree and should be excluded from the fence-line extraction process.

The algorithm uses a circular kernel around each point and checks the coverage within the circle and excludes circles that have a high coverage. This is based on the assumption that fence-lines do not have a large amount of coverage when considering circular sub-regions. This greatly reduced the computation needed to detect line segments but because of the radius and coverage threshold selected for the tree reduction filtering, smaller bushes still remain after filtering. Points on the larger tree are removed, however this same algorithm has been found to remove points at the intersection of fences.

5.2.2 Fence-line detection using Integrated Imagery and Lidar

Many of the steps developed for Lidar was used in the integrated process using both Imagery and Lidar, especially after the point clusters were created to begin the process of line segmentation.

The imagery was used to detect all the edges using Canny edge detection algorithm. As image processing is computationally intensive, small tiles of 256 pixels was used. This process created edges for everything in the image and was much more than necessary. Clusters of points were identified based on adjacency and vegetation edges were removed using curve calculated for each such cluster. RANSAC line fitting algorithm clustered line fits based on a threshold inlier value, and the results were exported as vectors which were filtered initially based on Lidar relative elevation and probabilistically by SVM.

In the results below from Figure 5-1 to Figure 5-6, discusses the various issues with fence-line extraction using imagery. The extracted line segments are shown in blue on the right and overlaid on the original imagery on the left in the figures below, while the Canny edge detector output is shown in the centre image as white pixels below.



Figure 5-1: Trees causing many edges, but no straight line edges

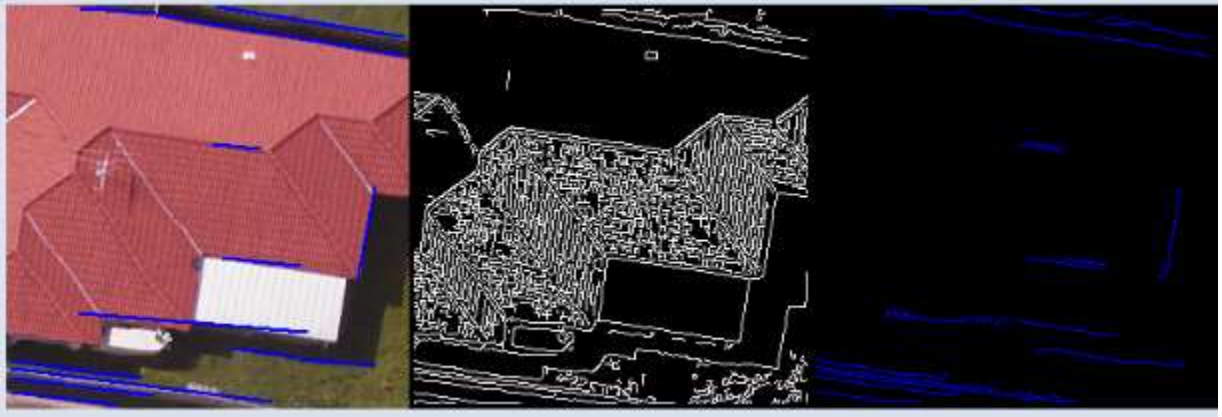


Figure 5-2: Roof texture causing many edges



Figure 5-3: Shadows and roads are detected in this case

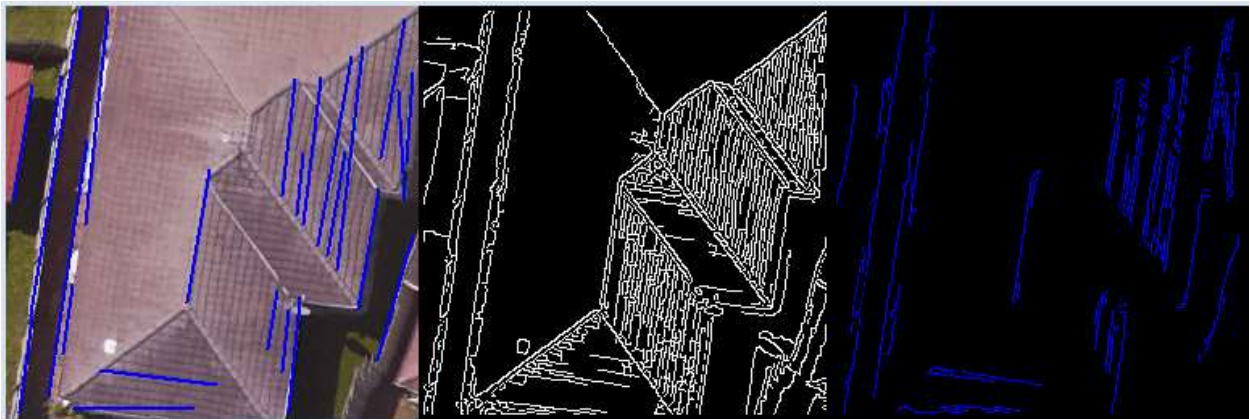


Figure 5-4: While the fence on the left is correctly identified, the algorithm also finds many other lines on the building

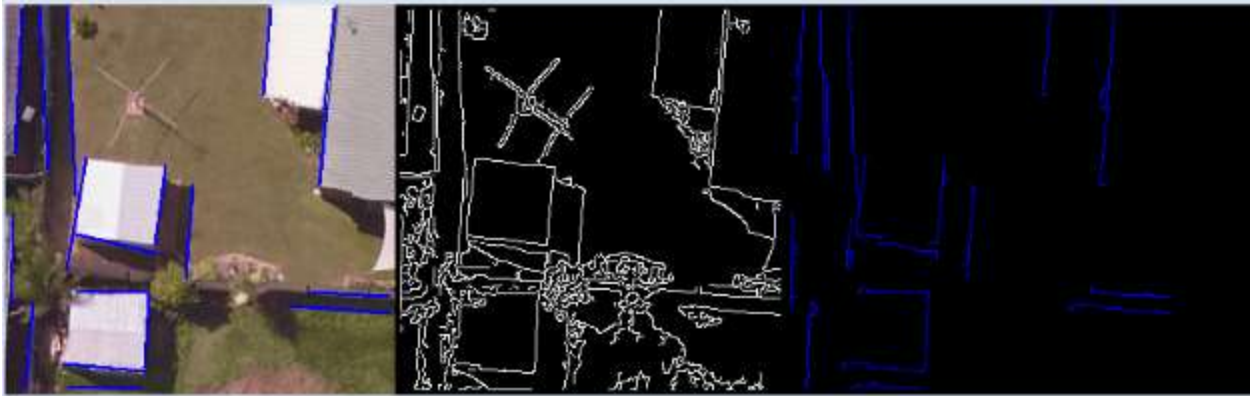


Figure 5-5: More buildings as well as some fence lines are detected here



Figure 5-6: SVM filtering results in lines in green to be selected as fences, and lines in red are rejected

Discussion: This approach extracts fence-line segments effectively, but unfortunately, it also extracts an even greater amount of other linear features in the imagery (Figure 5-6). For the Integrated Imagery-Lidar based approach a different SVM model was chosen, but in this case, the same attributes are not readily available, as there is no height information at pixel level. The level of accuracy with this type of classifier depends heavily on the amount of training data, and the more diverse the appearance of a fence line is, the more data is needed to successfully train a classification model.

5.2.3 Quality Assessment Tool for Fence-line extraction

A quality assessment tool was developed to evaluate the results of the fence-line extraction. The tool initially loads the fence-lines for visual quality assessment of the fence-line extraction. It is also used for SVM training and the manually edited fence-lines can be used as filtered output. Figure 5-7 shows the various options explored for using background image for visual assessment. The top-left image is the intensity, the top-right the DSM, the bottom left is the Lidar classified image and the bottom right is the hill-shading of the Lidar.

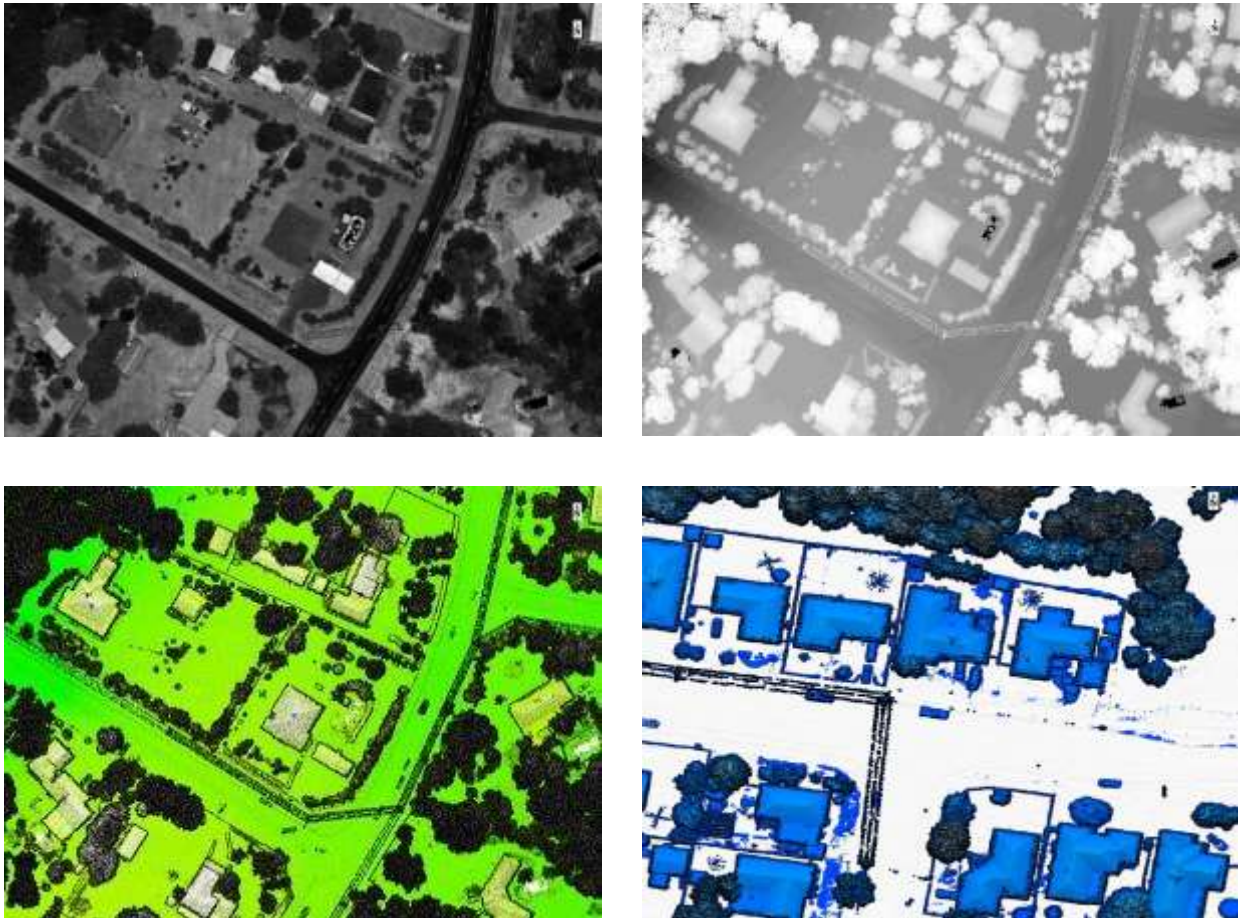


Figure 5-7: Various options explored for background imagery to be used in visual quality assessment, manual editing and SVM training

The hill-shaded (topographic shaded) raster representation created from the Lidar data can be useful for manual inspection of fence-line detections made by the algorithm. This is especially

true if imagery is not available, or if imagery is not from the same flight as the Lidar data. In cases where imagery was not collected at the same time, it could be impossible to know if a car was present or if a new building was constructed. Using the hillshade representation to inspect fence features can be very useful in these cases. The hillshade representation has a clear advantage over a height raster, in that fence lines tend to display clearly unless obstructed by vegetation. Figure 5-8 shows the background used for non-ENVI version and fence-line extract overlaid on imagery.

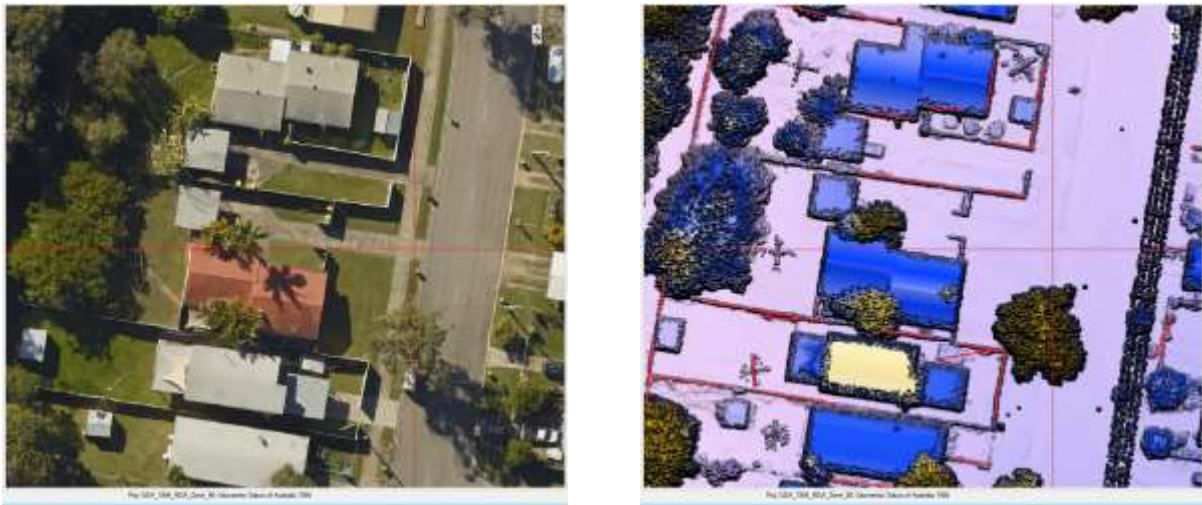


Figure 5-8: (Left) Aerial imagery background for non-ENVI version and (Right) fence-line overlay

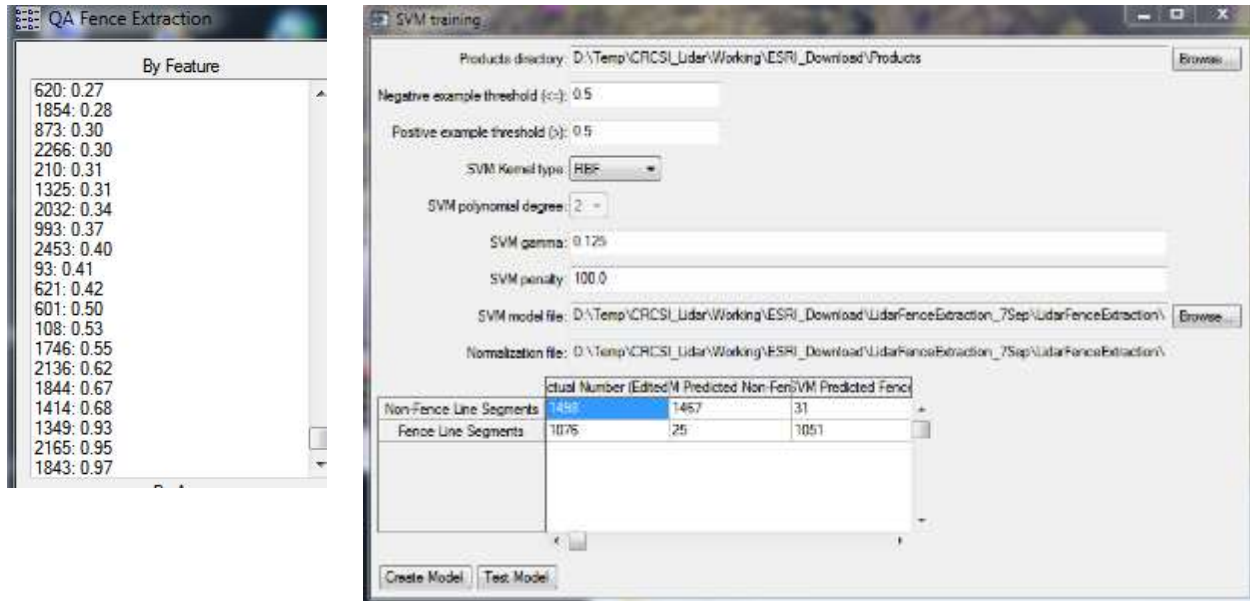


Figure 5-9: (Left) Probability values shown for each line segment; and (Right) SVM training model creation and accuracy reporting

The left image in Figure 5-9 above shows the probability values displayed for each line segment which changes to 0 if the line is chosen to be non-fence and changes to 1 if the line is selected as a fence. The image on the right shows the SVM model training and creation window and it also reports on the accuracy of the fence-line extraction process.

5.2.4 SVM implementation in ENVI

This section is adapted from Harris Geospatial website for implementing a SVM in ENVI proprietary software (<http://www.harrisgeospatial.com/docs/BackgroundSVM.html>).

Support Vector Machine (SVM) is a supervised classification method derived from statistical learning theory that often yields good classification results from complex and noisy data. It separates the classes with a decision surface that maximizes the margin between the classes. The surface is often called the optimal hyperplane, and the data points closest to the hyperplane are called support vectors. The support vectors are the critical elements of the training set.

The following outlines the steps for implementing a SVM in ENVI software with soft margin and uses a pairwise classification strategy for bilinear classification:

- Select the **Kernel Type** (different kernels have different options). The mathematical representation of each kernel is listed below:

<i>Linear</i>	$K(x_i, x_j) = x_i^T x_j$
<i>Polynomial</i>	$K(x_i, x_j) = (g x_i^T x_j + r)^d, g > 0$
<i>RBF</i>	$K(x_i, x_j) = \exp(-g \ x_i - x_j\ ^2), g > 0$
<i>Sigmoid</i>	$K(x_i, x_j) = \tanh(g x_i^T x_j + r)$

Where: **g** is the gamma term in the kernel function for all kernel types except linear; **d** is the polynomial degree term in the kernel function for the polynomial kernel; and **r** is the bias term in the kernel function for the polynomial and sigmoid kernels.

If the Kernel Type is **Polynomial**, set the **Degree of Kernel Polynomial** (range 1-6, default = 2). For a **Polynomial** or **Sigmoid** kernel type, specify the Bias in Kernel Function (default = 1.00, which is the "r" term in the above kernel functions). For a Polynomial, or **Radial Basis Function**, or Sigmoid kernel type, set the Gamma in Kernel Function to 0.01 (the "g" term used in the above kernel functions, default = inverse of the number of computed attributes).

- Specify the Penalty Parameter for the SVM algorithm to use. This value is a floating-point greater than 0.01, the default is 100.0. The penalty parameter prevents over-fitting and allows a degree of misclassification, which is important for non-separable training sets.
- Select a Threshold value to indicate level of confidence that the closest segments of any given class represent the same class as that segment.

5.2.5 SVM implementation in the GUI

This section details the SVM implementation in the GUI tool and the workflow. Specific filenames have been left as is so that it is easier to relate to the actual implementation when running the processes.

It is important to select clear training examples when creating a SVM model. It could be a good strategy when creating a new model to not use a prediction model at all when running the processing. This way all line segments start out as 0.5 probability. The segments can then be marked with only the segments that is required to use as examples for fence/non-fence (clear examples). After marking a roughly even amount of examples of fence/non-fence, the probability thresholds are set to 0.1, 0.9 respectively in the GUI to omit the unmarked examples (0.5).

The SVM is implemented after points in a cluster are converted to a line-segment using a RANSAC line fit algorithm. Each line segment is assigned an attribute vector consisting of ten attributes. The attribute vector is used as input to the SVM prediction model. The attributes are computed from the following:

- i. Length in X-Y
- ii. Linear least squares fit coefficient (along best RANSAC line)
- iii. Linear least squares fit coefficient (perpendicular to best RANSAC line)
- iv. Chi Square error metric of linear least square fit of original points.
- v. Covariance matrix of linear fit of original points (4 coefficients)
- vi. Z-component of plane fit normal vector (plane fit to original points)
- vii. Percentage of points within tolerance threshold of plane fit (low for vegetation, high for flat structures)

The feature vectors are stored in "FenceFeatures.tif" which is a [10, NumLineSegments] matrix. The points in each individual line segment are stored in "FenceGeometry.bin". The probabilities as predicted by the SVM model are stored in "FencesProb.float" which consists of

NumLineSegments 4-byte floating point values. If this file does not exist at the time the “Review” tool is started, then the review tool will create the file and populate it with all (NumLineSegments) 0.5 values. After each edit made in the review tool, this file is updated with the corresponding probability (1.0/0.0).

When creating a new SVM model this file is used as the input, and thresholds can be set to include a wider or narrower range of probabilities when selecting features. The SVM model generation then reads the “FenceFeatures.tif” and selects the subset based on the probability range selected.

The process of SVM model creation results in two files that describes how to apply the SVM prediction model to future processing runs. The first file is the scaling (or normalization) file. It is a TIFF file containing [10, 100] containing ordered (percentile) values for each attribute across the training data set used to train the model. It is recommended to have at least 100 examples when creating a model, generally, the more the better, but each example line segment should be carefully selected knowing that there aren’t any actual fence segment in the red group, or any non-fence segments marked green. The other file is a .model file which contains the SVM weights and biases used internally. The SVM library that is called from IDL passes in the model file name when computing the predictions from the normalized attribute vectors.

5.2.6 Implementation Versions of the GUI

One of the key consideration for this project is to develop a software product for fence-line extraction that can be used without any dependency on proprietary software. In order to achieve this it was proposed to have two versions of the software from an end-user point of view:

- **ENVI IDL based Software Version:** This version is designed and implemented for users that either own or have access to ENVI and ENVI Feature Extraction Licenses. It also requires a runtime version of IDL. The major advantage to the end-users is that it provides complete capability of the COTS product and provides an IDL-based

development environment for future enhancements and provides complete GUI based ease of use with minimal input from the user. ENVI and ENVI FX license are used for file access, DEM (bare-earth) raster creation, building, vegetation, and power-line classification. This means that a set of raw (unclassified) LAS files can be provided in conjunction with a shapefile containing processing polygons as input for processing while these would have to be created manually for the open source IDL version.

- **IDL based Software Version:** This is the open source implementation of the software. The IDL Virtual Machine (IDL VM) is a runtime version of IDL that can execute IDL '.sav' files without an IDL license. It runs on all IDL-supported platforms and does not require a license to run. IDL Virtual machine is freely downloadable product and runs the .sav file that is developed as part of this implementation. The key difference with the ENVI based software version is that, when ENVI is not licensed, the application behaves differently. It expects one or multiple classified LAS files. The majority of processing steps remain the same and it is expected that users will make use of any open source software to classify the LAS files to generate the classes required as input to this install.

Details of IDL VM is provided in the following link:

<https://www.harrisgeospatial.com/Support/SelfHelpTools/HelpArticles/HelpArticles-Detail/TabId/2718/ArtMID/10220/ArticleID/17309/The-IDL-Virtual-Machine.aspx>

5.3 Assessment of Algorithm for Fence-line Extraction

5.3.1 Assessing the algorithm and data for rural areas

The assessment of the algorithm performance in urban areas is done in Section 5.5 as part of the review of various Lidar resolutions. This section reviews the performance of the algorithm in rural areas. The rural area of Toowoomba had sparse Lidar data of 2-3ppsm so the parameter .json file was chosen accordingly. The fence characteristics of rural areas are usually chain-wire fences or posts or rail on posts (Figure 5-10). The algorithm has performed well for fence-line detection in rural areas (Figure 5-11 and Figure 5-12).



Figure 5-10: Example fence image in rural areas

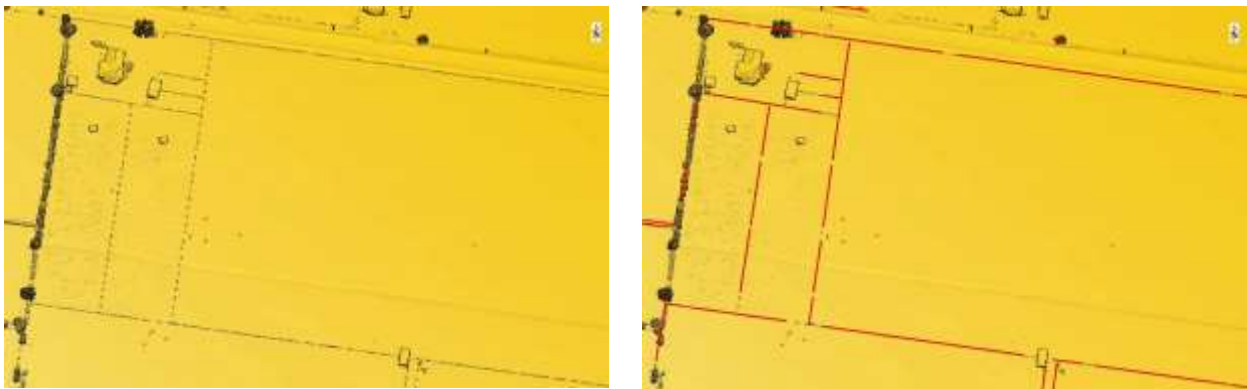


Figure 5-11: (Left) Lidar on fences and (Right) fence-line extraction



Another example area



Detected fence-lines before filtering



Fence-lines after SVM filtering



Example of SVM filtering (blue lines filtered out)

Figure 5-12: Example of rural fence-line extraction and SVM filtering

5.3.2 Assessing the feature extraction for Geiger Mode Lidar

A scene of Geiger mode Lidar was made available to the project for a location in USA. The site is 558m x 155m in the east coast of USA. Figure 5-13 left show the study area and right shows the point-cloud visualisation and the portion of the scene for which fence-lines were detected. The Geiger Mode Lidar was of 32ppsm, Figure 5-14 shows the fence-lines extracted from this type of Lidar. While the scene was too small and the fences of a different characteristics to the one in the parameter file, Geiger Mode Lidar provides a good option for fence-line detection.



Figure 5-13: (Left) Study Area in USA; and (Right) Geiger Mode Lidar point cloud



Figure 5-14: (Left) Unfiltered fence-lines and (Right) Filtered fence-lines extracted from Geiger Mode Lidar

5.3.3 Assessing the algorithm and data for Adelaide

Lidar data, Imagery and DCDB extract was made available for testing for Adelaide, South Australia. The Lidar data was of around 20ppsm and was reported to be captured at the same time as the aerial ortho-rectified imagery. The digital cadastral data from Adelaide had accuracy codes provided in it which was interpreted by accessing the South Australian government departmental website. Figure 5-15 shows the fence-line extract using the algorithm and medium density parameters.



Figure 5-15: Adelaide Lidar processing (Top) Unfiltered fence-lines (Bottom) Filtered lines

From visual inspection of the extracted fence-lines, it could be seen that the extract was not satisfactory. Further, close-up views in Figure 5-16 and Figure 5-17 reveal that regardless of the reported accuracy of the digital cadastre, the fence-line extract never matched the DCDB but rather matched the fences visible on the image. The image appeared to have been orthorectified very well using possibly the same Lidar and control points, yet it never matched the digital cadastre regardless of the accuracy of the cadastre. The only conclusion that can be drawn is that there is a systematic shift between the Lidar and Imagery capture with the digital cadastre. The reasons or magnitude of the shift were not explored to limit the scope of the project.



Figure 5-16: DCDB metadata states high accuracy but Lidar fence-lines match imagery not DCDB



Figure 5-17: DCDB metadata states low accuracy, still Lidar fence-line extract matches imagery not DCDB

5.4 Assessment of feasibility of DCDB upgrade using Fence-lines

After the extraction of the fence-line it was necessary to explore whether these lines could be used to adjust the digital cadastre. While it was necessary to test the digital cadastre where the geo-positional accuracy was already very low, the issue was the availability of Lidar in those areas. Similarly, where Lidar was available the geo-positional accuracy of the digital cadastre was very high.

In Morayfield the reported geo-positional accuracy of the digital cadastre was about 0.1m in most areas. It was therefore decided to distort the digital cadastre for Morayfield to simulate a low accuracy cadastre. The distortion was in rotation, translation and scale of around 63m which is the highest geo-positional inaccuracy in the DCDB in Queensland at the moment.

Figure 5-18 through to Figure 5-21 demonstrated that it was possible to block adjust the distorted cadastre based on the extracted fence-lines. Accuracy aspects and the time taken are explored in later chapters.



Figure 5-18: Simulated distortion of the digital cadastre



Figure 5-19: Link lines between extracted fence-lines and distorted cadastre for rubber-sheeting

FID	Shape *	Direction	Distance
4	Polyline	56-28-30	37.138
5	Polyline	52-13-43	37.864
6	Polyline	39-13-53	41.24
7	Polyline	36-39-8	42.299
3	Polyline	64-21-2	43.933
2	Polyline	71-41-34	44.908
8	Polyline	24-58-7	48.785
1	Polyline	76-0-47	49.733
9	Polyline	27-54-50	52.111
10	Polyline	24-52-14	58.483
0	Polyline	76-24-29	61.824
11	Polyline	25-35-55	62.42
17	Polyline	70-38-48	67.756
20	Polyline	48-42-51	68.722
12	Polyline	25-24-44	69.779
19	Polyline	44-30-59	72.51
13	Polyline	26-58-7	72.57
16	Polyline	57-49-35	75.425
18	Polyline	39-8-33	76.351
14	Polyline	30-19-51	77.643
15	Polyline	49-36-4	82.616

Navigation: 1 (0 out of 21 Selected)

Figure 5-20: Variable distance and direction of the link lines due to rotation, translation and scaling of the distorted cadastre



Figure 5-21: The DCDB was able to be block adjusted to its original high accurate position using the extracted fence-lines

5.5 Assessment of Various Lidar Data Resolutions for Fence-lines

The assessment for suitability of various Lidar data resolution expressed as points per square metre for fence-line extraction was done as shown in Figure 5-22. The direction of capture was evaluated for two purposes, firstly to test if the orientation of the Lidar flight lines affected the number of points falling on the fence tops, and secondly to test whether the developed algorithm had a bias towards any direction of data capture.

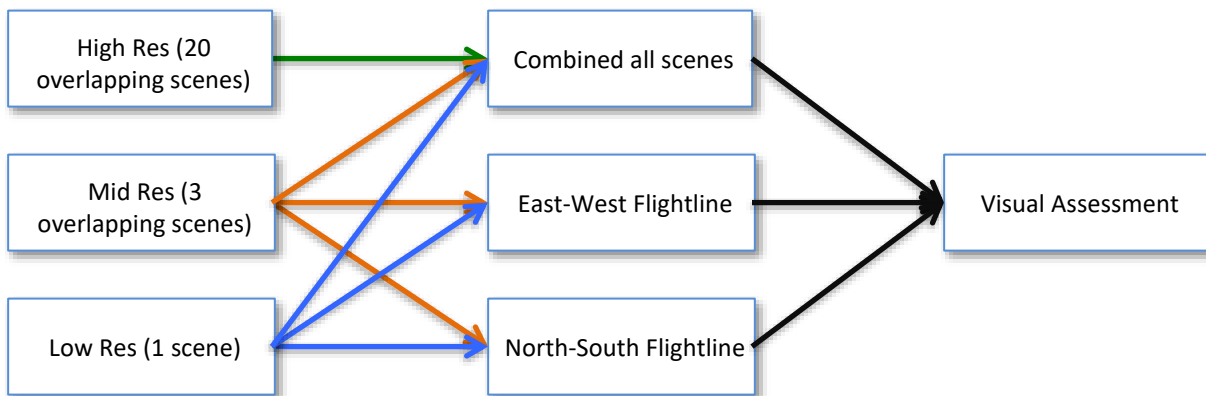


Figure 5-22: Assessment of various Lidar resolutions and capture directions for fence-line extraction

5.5.1 Evaluation of high resolution Lidar

Twenty overlapping Lidar scenes on the southeast corner of the Morayfield area of interest were selected. As each scene is 8ppsm, so the resulting average density of Lidar was around 160 in many parts of the scene. As the point density was high, the direction of the flight lines was not evaluated for the high-resolution composite scenes. Filtered fence-lines were extracted using high-resolution parameters and default SVM that was then overlaid on imagery to examine the number of fence-lines detected, and their errors of commission and omission. Figure 5-23 shows that visible fence-lines have been extracted for most of the features in the area.



Figure 5-23: Fence-line extraction from high-resolution extraction shows a large percentage of visible fence-lines extracted

5.5.2 Evaluation of medium resolution Lidar

For this evaluation three scenes each in the vertical and horizontal flight direction were used and a final combined extraction performed for all six images together.





Figure 5-24: Fence-lines extracted from medium resolution Lidar for (Top) Horizontal and (Middle) Vertical directions of flight; and (Bottom) the combined Lidar results

From Figure 5-24 it can be seen that medium resolution Lidar is capable of delivering similar results compared to high resolution Lidar however there are more errors of commission. From the (Top) image for horizontal flight-lines it can be seen that there are more horizontal fences extracted and similarly more vertical lines for vertical direction of capture for (Middle) image. The combined extraction (Bottom) shows both horizontal and vertical fence-lines captured, however there are still some errors of commission compared to the high resolution feature extraction.

5.5.3 Evaluation of low resolution Lidar

For this evaluation, one scene of 8ppsm in each of the two directions are considered low resolution and the combined feature extraction in the third stage serves to evaluate the effect

of direction on the data points falling on fences as well as the directional bias of the algorithm (Figure 5-25).





Figure 5-25: Fence-lines extracted from low resolution Lidar for (Top) Horizontal and (Middle) Vertical directions of flight; and (Bottom) the combined Lidar results

The results show that the low resolution Lidar data extracts less fence-lines compared to the previous two. There are some errors of commission and more errors of omission. The combination provides some added fence-lines that are extracted, however they are not adequate for a DCDB block adjustment.

5.6 Summary

The algorithm that was developed performs better than expected. The majority of the missing fence-lines are due to vegetation and small buildings close to a fence, more often a partial fence-line is detected, rather than a complete miss.

From the discussions in this chapter it was found that Lidar data should be collected with variations in flight pattern, point densities and different time of year for vegetation on or off. More experimentation is required with changing algorithm parameters to find optimal settings as well as adding other false positive reduction strategies, for example, specifically look for cars or other common objects.

6 Accuracy Achievable

6.1 Accuracy of Lidar Data

Lidar data was validated using ground truth obtained from GPS field survey using Lastools for data shown in (Table 6-1 Top). The results in (Table 6-1 Bottom) show that with an RMSE of 0.045, the Lidar data was within the vertical limits of accuracy as per the ASPRS Guidelines for Vertical Accuracy Reporting of Lidar Data (Flood 2004)

$$\text{Accuracy}_z = 1.96 * \text{RMSE} = 0.088,$$

Which is within limits of the recommended 0.150.

Table 6-1: Lidar data validation using GPS field survey coordinates

Stn.	GPS _{East}	GPS _{North}	GPS _Z	LIDAR _Z	Diff
PM160643	494144.660	7000577.050	53.779	53.746	-0.033
PM160753	494764.765	7001017.771	59.348	59.438	0.090
PM165154	494900.611	7000646.300	58.029	58.096	0.067
TGT1	494903.245	7000645.653	58.100	58.103	0.003
TGT2	494228.762	7001027.850	51.817	51.838	0.021
TGT3	494604.607	7001323.800	51.500	51.526	0.026
TGT4	494766.851	7000993.469	59.753	59.777	0.024
TGT5	494142.412	7000575.016	53.772	53.809	0.037

Avgabs	RMS	Stddev	Average	Skew
0.0377410	0.0459019	0.0376782	0.0294074	0.0331676

6.2 Accuracy of Filtering using SVM

A probability is computed for every extracted line segment using the default prediction model. Segments that are incorrectly classified can be edited (re-assigned) by simply clicking on the line segment. There are three states when clicking on a line segment: original probability (as displayed in Figure 5-9 left), assigned non-fence (0.0), and assigned fence (1.0).

When clicking on “Test Model” (lower right button in Figure 6-1), the table is updated with the predicted vs. labelled (actual) counts. In this case below, there are 459 line segments that are labelled non-fence, and 509 line segments that are labelled fence. The two rightmost columns in Figure 6-1 are defined as the Confusion Matrix.

The following statistics can be computed from the Confusion Matrix (Table 6-2 and Table 6-3).

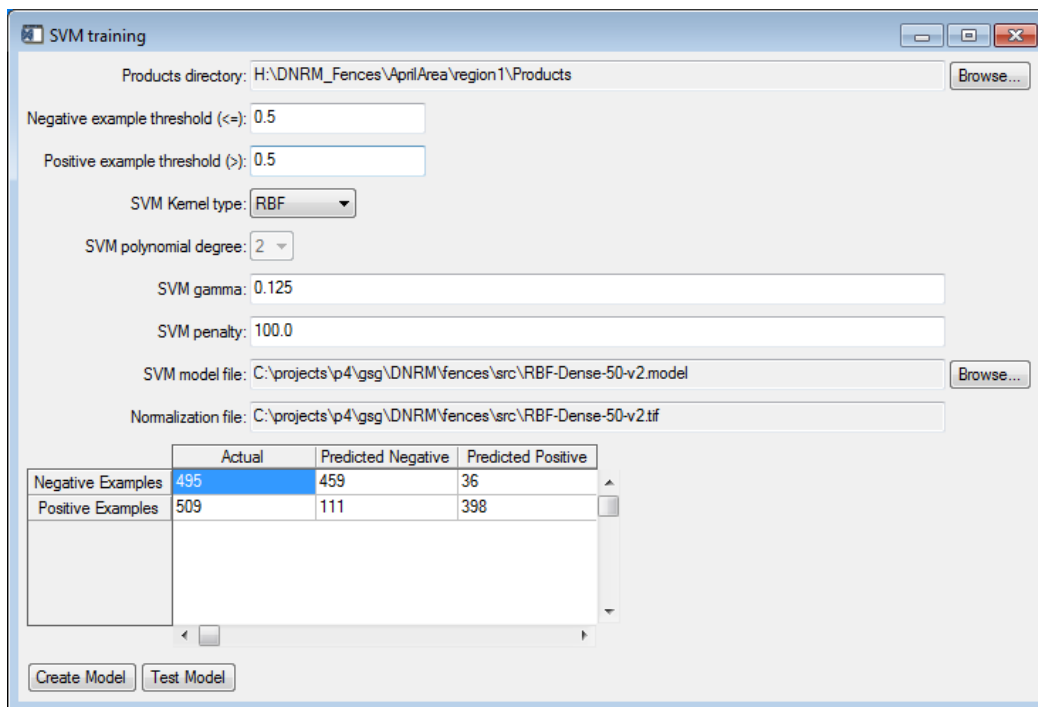


Figure 6-1: Example SVM Training Window

Table 6-2: Confusion Matrix from SVM Training

		Predicted	
		Non-Fence	Fence
Actual	Non-Fence	459 (TN)	36 (FN)
	Fence	111 (FP)	398 (TP)

Table 6-3: Classification metrics from the Confusion Matrix

Class 1 (non-fence)		Class 2 (Fence)	
True Negative (TN):	459	False Negative (FN):	36
False Positive (FP):	111	True Positive (TP):	398
Precision:	0.80526	Precision:	0.91705
Recall:	0.92727	Recall:	0.78193
F1 Score:	0.86197	F1 Score:	0.84411

Where:

True Positive (TP): When the SVM predicts a line is a fence, and it is an actual fence-line (Fence – Fence Matrix)

False Positive (FP): When the SVM predicts a line is a fence, but it is not a fence-line (Fence – Non-Fence Matrix)

True Negative (TN): When the SVM predicts a line is not a fence, and it is not a fence-line (Non-Fence – Non-Fence Matrix)

False Negative (FN): When the SVM predicts that the line is not a fence but it is actually a fence-line (Non-Fence – Fence Matrix)

Precision: When the SVM predicts that the line is a fence or a non-fence, this is a measure of how often it is done:

$$\frac{TN}{(TN+FP)} \text{ or } \frac{TP}{(FN+TP)}$$

Recall: When the lines are identified as a fence, how often is it an actual fence or when it is identified as non-fence how often is it not fence:

$$\text{True positive rate} = \frac{TP}{(FP+TP)}; \text{ True negative rate} = \frac{TN}{(TN+FN)}$$

F1 Score: A measure of a test’s accuracy, it is the harmonic mean of precision and recall:

$$2 \times \frac{\text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})}$$

The recall and precision statistics will vary depending on the following factors:

- A. The geographic area being evaluated, for example, a rural or forested area will give a different result than an urban or suburban area.
- B. The point density and quality of the Lidar data collection.
- C. The parameters used as input to the algorithm. As part of this effort, three sets of parameters were created targeting high, medium and low density Lidar collection. However, there is an opportunity to further refine individual algorithm parameters to obtain better results depending on the Lidar data characteristics.

Note that by changing algorithm parameters such as increasing the CLUSTER_DISTANCE, or increasing the PERPENDICULAR_TOLERANCE, the number of detections can be increased at the expense of also increasing false positive detections. This would have the effect of increasing the recall percentage and also decreasing the precision. On the other hand, decreasing the same parameter values would lead to increased precision at the expense of lowering the recall.

6.3 Accuracy of Extracted Fence-lines

The developed workflow has shown promising results, with extraction accuracy that allows for an accurate adjustment of the existing cadastre. Fence-lines extracted from Lidar have a combined horizontal accuracy of 0.282m (Table 6-4 bottom) while fence-lines extracted from imagery have a combined horizontal accuracy of 0.258m (Table 6-5 bottom). Although the accuracy of image-based feature extraction appears to be better, the number of fence-line segments extracted from Lidar is significantly higher in number.

Table 6-4: Computing the accuracy of Fence-line extraction from Lidar (Top and Bottom table)

X_{DCDB}	Y_{DCDB}	X_{Lidar_Fence}	Y_{Lidar_Fence}	X_{Error}	Y_{Error}
494379.764	7000980.742	494379.687	7000979.786	0.077	0.956
493672.230	7000754.308	493672.208	7000754.305	0.022	0.003
493714.640	7000687.245	493714.551	7000687.375	0.089	-0.130
493854.982	7000651.756	493854.896	7000651.368	0.086	0.388
494255.721	7000606.390	494255.761	7000606.279	-0.040	0.111
494279.534	7000602.681	494279.644	7000602.585	-0.110	0.096
494531.295	7000784.481	494531.388	7000784.400	-0.093	0.080
494488.371	7001390.521	494488.139	7001390.831	0.232	-0.309
494486.443	7001350.210	494485.988	7001350.145	0.456	0.065
494485.931	7001338.840	494485.555	7001338.787	0.376	0.052
494513.212	7001343.901	494513.232	7001343.899	-0.020	0.002
494484.893	7001316.867	494484.991	7001316.802	-0.098	0.065
494471.641	7001306.481	494471.550	7001306.356	0.091	0.126
494302.425	7001240.759	494302.432	7001240.654	-0.007	0.104
494276.420	7001121.264	494276.085	7001121.124	0.335	0.140
494635.551	7000995.760	494635.453	7000995.721	0.098	0.039
494044.599	7000915.606	494044.560	7000915.450	0.039	0.156
494585.267	7000754.505	494585.408	7000754.476	-0.141	0.029
494685.670	7000597.796	494685.414	7000597.907	0.256	-0.111
494445.236	7000797.994	494445.229	7000797.984	0.007	0.010
494422.577	7000731.933	494422.660	7000731.913	-0.083	0.021
493795.802	7000740.007	493795.726	7000739.939	0.076	0.067
494124.036	7000903.636	494123.960	7000903.587	0.076	0.049
493678.176	7000793.847	493678.235	7000794.026	-0.059	-0.179
494653.947	7000985.521	494653.886	7000985.694	0.061	-0.173
494701.900	7001035.506	494701.696	7001035.608	0.204	-0.102
494682.284	7000997.971	494682.330	7000997.826	-0.046	0.145

$RMSE_x$	$= \sqrt{RMSE_x^2 + RMSE_y^2}$	0.166
$RMSE_y$		0.228
Horizontal Accuracy	$= \sqrt{\frac{\sum_{i=1}^n (DCDB_i - Fence_i)^2}{n}}$	0.282

Table 6-5: Computing the accuracy of Fence-line extracted from Imagery & Lidar (Top and Bottom table)

X_{DCDB}	Y_{DCDB}	X_{Image_Fence}	Y_{Image_Fence}	X_{Error}	Y_{Error}
494524.142	7000773.804	494524.173	7000773.873	-0.032	-0.069
494531.295	7000784.481	494531.400	7000784.393	-0.105	0.088
494635.551	7000995.760	494635.243	7000995.855	0.308	-0.095
494585.267	7000754.505	494585.346	7000754.510	-0.078	-0.004
494544.389	7000777.202	494543.834	7000777.316	0.555	-0.113
494445.236	7000797.994	494445.267	7000798.039	-0.030	-0.045
494422.577	7000731.933	494422.645	7000731.825	-0.067	0.109
494653.947	7000985.521	494653.696	7000985.685	0.251	-0.164
494682.284	7000997.971	494682.164	7000998.107	0.120	-0.137

RMSE _x		0.237
RMSE _y	$= \sqrt{RMSE_x^2 + RMSE_y^2}$	0.102
Horizontal Accuracy	$= \sqrt{\frac{\sum_{i=1}^n (DCDB_i - Fence_i)^2}{n}}$	0.258

6.4 Sources of Error in Fence-line extraction

The fence extraction process is susceptible to a few different types of errors. Broadly, the categories of errors are (a) missed fences, (b) inaccurate line direction, and (c) false positive detections. This section discusses the errors in Lidar fence-line extraction process.

6.4.1 Missing Fence-lines

For missing fence lines, the most prevalent causes are small sheds/buildings or unusual shaped structures in the same elevation range that can lead to too many cluster of points confusing the line detection, thus causing a real fence line to be missed (Figure 6-2). Trees or significant vegetation close to fences can also have the same effect of confusing the line detection (Figure 6-3). Additionally due to the use of tree removal algorithm, fence intersection points can cause a similar problem in that there is too much coverage over a circle causing points to be removed from consideration (Figure 6-4).



Figure 6-2: Small building/shed (blue X) is causing a section of the fence to be missed



Figure 6-3: An example of fence line segments missed due to vegetation



Figure 6-4: Missing fence intersection lines because of circular coverage reduction of point cluster by the tree removal algorithm

6.4.2 Inaccurate Line Direction

The second category of error is due to inaccurate fence line direction normally caused by the vegetation resulting in the line fitting to be skewed towards the vegetation. As shown in Figure 6-5 the line represented in red over the fence is pointing towards the vegetation. This is prior to the final filtering step leading to the dashed red lines being removed.



Figure 6-5: Fence-line being skewed towards a cluster of vegetation

6.4.3 False Positives

The last category of errors is false positives. The most common causes of false positives are cars, vegetation, and to a lesser extent buildings. Vegetation misclassified as a fence line is more prevalent when processing lower density Lidar datasets. This type of misclassification is rare in the highest density Lidar test dataset. Figure 6-5, Figure 6-6 and Figure 6-7 show some examples of false positive errors.



Figure 6-6: An example of parked cars being misclassified as a fence line is shown here, (blue arrow)



Figure 6-7: An example of the sides of a building being misclassified as a fence line (not common in Lidar extraction but very common in image-based extraction)

6.5 Summary

This chapter discussed the accuracy aspects of the data input and data output, as well as the sources of error. Accuracy assessment of input Lidar data based on GPS field survey confirmed a published method of post data capture assessment of Lidar data as well as confirmed that the data capture for this project over Morayfield was of a very high vertical accuracy.

Next, accuracy parameters obtained from SVM training tool was discussed in detail with explanations on how to interpret the results reported by the SVM training tool.

Further, accuracy of fence-line extractions for features obtained from Lidar and imagery was evaluated and the comparison between the extracted lines and the existing DCDB was within RMSE of 0.282 and 0.258 for Lidar and Imagery extractions respectively. Thus, for areas where the fence-lines are a close approximation of the property boundary, this method can be used to upgrade the graphical representation of the digital cadastre with a high degree of correlation to the fences.

Finally, the sources of errors in fence-line extraction was discussed with various examples. The methods to minimise these errors were not discussed, but it needs a revisit of the variable parameters as well as the processing steps.

7 Recommendation for Cadastral Upgrade

7.1 Background

This chapter discusses the big picture of cadastral data upgrade and the considerations for the operational implementation by cadastral jurisdictions. Firstly, operational opportunities and benefits to custodians of digital cadastral data are explored based on time savings, accuracies obtained and how much effort would be needed to integrate it in the existing processes.

It then explores the limitations and expected issues in the implementation of the process. For CRCSI/FrontierSI it presents an opportunity to commercialise not just the process, but other learnings and prospects.

Finally, there is an in-depth analysis of time taken for fence-line extraction followed by recommendations on relative advantages and resolutions of Lidar and Imagery as well as algorithm improvement.

7.2 Operational Opportunities

7.2.1 Potential Benefits to Organisations

The developed process and the GUI have provided a start to organisations looking to implement it. The output from this research project has proved that it is feasible to upgrade the digital cadastre using the developed algorithm.

There are multiple other considerations such as whether the accuracy of the existing cadastre is already greater than that could be attained by this process, or whether there are other processes (planned or implemented) for improvement of cadastral upgrade that are already in the pipeline, or whether there is time and resource impact from implementing this process.

7.2.2 Time savings

An analysis done in Section 7.5 below shows that there are significant time savings by using this process. Although this analysis displays data for an area of 1km x 1km, automation is possible by this process where the entire feature extraction for a given area can be processed in one run.

The algorithm has a further fail-safe mechanism where if a process is interrupted in the middle due to any issues, the algorithm can re-start at the last processing point and complete the extraction.

A further advantage of the algorithm is that the processing is done in tiles and the output are in individual folders named as per the tiles, so cadastral upgrade processes can start using the tiles already processed, even when the fence-line extraction process is continuing to run.

7.2.3 Accuracy Attained

Accuracy analysis of various aspects of the output of the project was done in detail in chapter 6. It was seen that for the area studied, the horizontal RMSE using Lidar data is 0.282. In DNRME, the accuracy attribute of the digital cadastre for processes using ortho-imagery are reported between 2m to 5m. The accuracy obtained from the fence-line extraction in this project has exceeded that accuracy while maintaining a higher speed of data upgrade.

7.2.4 Process Integration

The fence-line extraction produces a shapefile that can be integrated in most GIS software. This provides an opportunity of integrating current data upgrade processes by using fence-lines as the input for digital cadastral block adjustment.

7.3 Limitations

7.3.1 Accuracy Expectations

While the previous section has shown that there are clear advantages of implementing the algorithm for cadastral upgrade, there are some considerations and limitations that need to be taken into account.

The accuracy attained from this process have been for ideal conditions and may be different when using it in a real-world situation. The accuracies attained are dependent on data, topography, fences being in the correct location etc.

Also, these are not a replacement to survey accurate data and can only be used as an interim measure to improve the geo-positional accuracy of the cadastre while waiting for other processes to improve the accuracy.

This process will however improve the “look” of the data for landowners viewing their cadastral boundary lines in online portals and search engines, where there will be no visible difference in the boundary lines and the fences.

7.3.2 Time Expectations

The algorithm was run in an 8 Core Workstation which would have improved the speed of processing significantly. There would be other variables for speed of processing such as data density, and whether there was some additional data clean-up necessary for improving the results of the fence-line extraction.

This can however be addressed by having a dedicated powerful computer just for processing this algorithm while the output would be used in other existing computers.

7.3.3 Ease of Adaptation of the Algorithm

The algorithm was developed progressively and the GUI and other specific processing steps were developed in conjunction, therefore the steps and processes have become easy for the project team to understand and implement which might be difficult for a fresh user to learn.

Also, the algorithm was developed as two versions, one open source IDL and the other ENVI IDL, and the latter version has more functionalities. This means that users with no ENVI licence would be able to run the program with no difficulty but they would not have all the ease of use as the licensed version.

The GUI is however easy to use with all menu items one-level deep and accessible with intuitive steps and users can familiarise with the process relatively easily. There is a step by step instruction on how to run the GUI in Appendix 10.7.

7.3.4 Data Availability

The biggest issue with using this process is the availability of Lidar data. The imagery part of the algorithm still needs a significant amount of work before it can be considered as useful as the Lidar algorithm. Lidar data is expensive to capture and it has been shown in Section 5.5 that at least a medium density Lidar of more than 20ppsm is needed, which is an expensive initial outlay. For larger states this becomes a much bigger issue with costs for Lidar data capture.

Geiger Mode Lidar however provides a cost effective method of capturing large areas Lidar data. With a coverage of 1000 to 1500 square kilometres per hour, it can capture large areas of data for similar weather and climate conditions in one data capture mission. This Lidar has shown to provide good results for fence-line extraction. An added benefit of a Lidar capture, whether Geiger Mode or existing aerial Lidar, is that these data have multiple uses such as elevation models, contours, city models, flood modelling, monitoring etc.

7.3.5 Existing processes

While the fence-line extraction process is seen to fit with existing upgrade process of Queensland, it is not certain how it will fit with other cadastral jurisdictions. There is also the possible human element of rejecting changes in existing processes among DCDB operators.

The most likely solution to this is firstly to test if this process fits existing processes and secondly to train cadastral upgrade staff.

7.4 Commercialisation Opportunities for CRCSI

The source code in IDL for the GUI can be re-packaged in other languages such as Python if required. The open source IDL itself can be used to modify aspects of the GUI and other feature extraction code for other purposes.

There is further opportunity to develop an enhanced add-on package for users of a licensed package such as ENVI which is independent to the original package but acts as an add-on subscription based package (e.g. SurvaCAD is an add-on package on AutoCAD).

The developed workflow and GUI has the potential to be used in other feature extraction processes apart from fence-line detection by altering parameters, so there are further opportunities to package this for a broader use in the field of feature extraction, Lidar processing, and image processing.

As there is now a substantial body of work and understanding of the various processes, data manipulation, Lidar processing, image-processing, accuracy assessments etc., these can be used as commercialisation opportunities by providing training and consulting services.

There can be further opportunities for commercialisation by publishing books or training materials that are a direct outcome of this project and which can be enhanced by tailoring it to the theme of a book or training material to be developed.

Finally, since there is a great need for upgrading the digital cadastre for cadastral jurisdictions and there have been examples of this kind of work being outsourced, there might be commercialisation opportunities for CRCSI/FrontierSI of facilitating the upgrade of the digital cadastre for other states and territories in Australia and internationally.

7.5 Processing Times for Fence-line extraction

One consideration for using this method for cadastral upgrade is to ascertain whether it provides actual time savings. One of the main problems with the current data upgrade method was the time and labour intensive processes.

Table 7-1: Time taken for fence-line extraction from various Lidar data sources and resolutions

Data Density	# points	Area coverage	Total Time (a+b+c)	Project creation (a)	Classification (b)	Fence extraction (c)
Low Res Vertical	761,316	300m x 301m	3 min	1 min	1 min	1 min
Low Res Horizontal	946,131	300m x 301m	3 min	1 min	1 min	1 min
Low Res Combined	2,952,146	500m x 501m	3 min	1 min	1 min	1 min
Mid Res Vertical	13,145,120	970m x 1001m	12 min	1 min	2 min	9 min
Mid Res Horizontal	6,585,071	500m x 501m	8 min	1 min	2 min	5 min
Mid Res Combined	25,580,305	1000m x 1001m	24 min	2 min	5 min	17 min
High Res Combined	63,829,555	1000m x 1001m	41 min	4 min	18 min	19 min
Geiger Mode	397,472	99m x 97m	1.5 min	0.5 min	0.5 min	0.5 min
Adelaide	14,177,031	1000m x 1001m	15 min	1 min	2 min	12 min

Table 7-2: Time calculation for fence-line extraction in another computer at Harris Geospatial, USA

Location	# points	Total Time (a+b+c)	Project creation (a)	Classification (b)	Fence extraction (c)
Test Area 1	62,947,783	95 min	40 min	21 min	34 min
Test Area 2	71,947,201	105 min	39 min	28 min	38 min

Table 7-1 shows data from an evaluation of Lidar over the project area of Morayfield and additional areas and Lidar such as Geiger Mode and Adelaide. The flight lines for this capture was both East-West (horizontal) and North-South (vertical). The Low res (resolution) data represented one scene of 8ppsm, the Mid res represented three scenes of 24ppsm, and High res represented 20 scenes.

It can be seen that the most time taken for fence-line extraction for a 1000m x 1000m area was 41 minutes for the highest density Lidar. In a similar area for medium resolution Lidar, the maximum time taken was 24 minutes.

To eliminate the possibility that one computer might be processing the algorithm quicker, the same algorithm was tested in another computer in Harris Geospatial, USA, although the specification of the computer used is unknown. Table 7-2 shows that for a very large number of Lidar point clouds, the processing time was a little over an hour and a half.

With time allowance for creating link files for block rubber sheeting of DCDB lines, the entire process could be completed within an hour to two hours. In speaking with DCDB upgrade staff at DNRME, this process could be expected to take a week to two weeks depending upon the complexity of the area and data available. Therefore, this process does provide significant time savings for cadastral jurisdictions looking to upgrade the cadastre as an interim step towards numerical cadastre, whenever that may be achieved.

7.6 Lidar and Imagery Capture – Relative Advantages

When comparing the use of imagery to the use of Lidar data to extract fence lines, Lidar data has some significant advantages.

Lidar has a true 3D point representation which gives the ability to use the height information. Perspective (look angle) does not impact the positional accuracy of the points for Lidar data capture compared to imagery. Sometimes it can be difficult to distinguish the top from the bottom of a fence at an oblique angle (depending on spatial resolution).

There are existing Lidar algorithms for extracting terrain and buildings as a pre-processing step. Lidar can be flown at night or during the day, and is not affected by shadows while imagery needs sunlight, and shadows depending on the time of day can create linear features in the imagery that can be confusing to a line extraction algorithm.

Imagery needs to be ortho-rectified which can introduce errors, however, imagery is typically less expensive to collect and can be collected more quickly. Also, imagery can be collected at a very good resolution which allows for easier interpretation by a human analyst.

7.7 Lidar and Imagery – Recommended Capture Resolution

While high resolution Lidar provides better fence-line extraction and less errors of commission, the mid resolution Lidar of around 20-30ppsm seem adequate, which translates to 3-4 overlapping Lidar scenes.

For imagery, the Morayfield scenes were captured at 6cm GSD, while the ortho-rectified imagery supplied for Toowoomba was 10cm GSD. While there was no appreciable difference between the two, it is recommended that either the ortho-rectification is done in-house using highly accurate control marks and DEM, supplied data could be checked rigorously for accuracy and fit before used for processing or validation.

From the evaluation in Section 5.5, it could be seen that two different directions of Lidar data capture assisted in covering fences in all directions, however, it is not clear if the same holds for imagery capture. It was also not clear if the algorithm had a bias towards any direction of data capture.

7.8 Algorithm Improvement

There have been some sources of error identified in Section 6.4, and the algorithm processing steps are listed in Section 5.2. Also the source code of the project is made available. This provides an opportunity to improve the algorithm to reduce the errors and false positive detections. There is also opportunity to improve the imagery based fence-extraction algorithm.

7.9 Summary

From evaluation of the opportunities and limitations of the algorithm and the time and labour savings that this can provide it is clear that organisations can benefit from utilising this process. There are multiple things to consider before taking such as step which have been discussed in this chapter.

8 Conclusion and Future Research

8.1 Review of Aim and Objectives

As discussed in Chapter 1, the problem identified with the digital cadastre of Queensland was that there was a large variation in the positional accuracy of the graphical representation of the cadastre in the DCDB (Appendix 10.1).

Further, this dataset was used by many organisations to link their spatial dataset or asset management systems which caused issues for the respective organisations because most of them had their asset location at a better spatial accuracy.

The problem with cadastral data management authorities like the DNRME in Queensland was that the existing processes of upgrading the cadastral data to a better spatial accuracy had to be done using manual processes and aerial imagery in conjunction with the update of the database from surveyed data for new development that was being submitted to the department by surveyors, which meant many surveyed data had to fit the existing inaccurate representation.

There is an identified need to explore different ways to speed up the process of upgrading the digital database. Thus, *the aim of this project was to evaluate the feasibility of utilising Lidar and Imagery to extract fence-lines for geo-positional upgrade of digital cadastre data and to evaluate the accuracies obtained.*

To achieve the aim of the research, four research objectives were defined. The objectives and their associated research questions assisted to guide this research project and the results of each of the objectives are discussed in the subsequent sections below (Sections 8.2 to 8.5).

8.2 Objective 1: Upgrade Methodologies

The first objective was to *“develop upgrading methodologies for cadastral data based on automated feature extraction and to assess their applicability and potential for operational implementation by partner land agencies”*.

A two-step strategy to deliver an upgrade methodology was formulated. Firstly, various methods of fence-line extraction were explored; and next, the extracted fence-lines were used to block shift and adjust the cadastre that included evaluation of accuracy metrics in each stage.

The primary plan was to extract fence-lines for the upgrade methodology. Various methods of feature extraction were explored and finally a graphical user interface developed using various semi-automated steps for feature extraction was selected.

Further experimentation was done to explore the usability of imagery alone. An edge detection algorithm was used, however this produced a large number of lines that were considered noise, as it was not possible to distinguish fence-lines from other lines. The process used in conjunction with Lidar to filter out lines not falling within a certain elevation range produced a better result, however, as fences in imagery are not orthogonally projected and are reliant on look angles, this produces lines that are not vertically in a position that the footprint of the fence would be. Further investigation perhaps by using convolution neural network is required for this process.

The objective of developing an upgrade methodology was achieved and it has been discussed in detail in Chapters 4 of this document.

8.3 Objective 2: Evaluation of Data Sources

The second objective of this research project was to *“evaluate the contribution of remotely sensed data sources (e.g. airborne and satellite imagery, and Lidar) to upgrading the spatial accuracy of the digital cadastre”*.

The contribution of airborne Lidar and Imagery to achieve the aim were evaluated and is discussed in Chapter 5. The existing processes used in DNRME utilised ortho-rectified imagery to upgrade the cadastral data to a better spatial location by block-shifting the digital data to identified fence-lines in the imagery.

There has been a significant investment made by the department in the current processes to customise the steps necessary to upgrade the cadastral data, and existing staff are familiar with the processes. Unless there are significant changes made to the current processes, it was logical therefore that the same processes would be continued to be used and any improvement made in automated identification of the fence-lines would assist to speed up the process of block-shifting the cadastral lines and snapping to the identified fence-lines through a semi/fully automatic process.

Evaluation made of the aerial Lidar and imagery demonstrated that they were capable of identifying fence-lines at varying resolutions of the data. Further evaluations were made of the various resolutions of Lidar data that would provide an optimal solution to extracting fence-lines from the point cloud. Finally, it was concluded that processes using Lidar, and to some extent integrated Imagery and Lidar processing, was capable of contributing to the upgrade of the spatial accuracy of the digital cadastre.

8.4 Objective 3: Accuracy Achievable

The third objective of this project was to *“identify, through experimental testing, the accuracy achievable from those data sources individually and in combination”*.

Accuracy testing of the data and the results were done for multiple aspects and reported in Chapter 6. Initially it was necessary to determine the accuracies of the Lidar data and the aerial

imagery. After the evaluation of the accuracies of the input data, evaluation of the accuracies obtained from the various processing outputs were planned.

Initial evaluations of the ortho-rectification done for the aerial imagery captured over the Morayfield area revealed a systematic shift of 1.2m with respect to lines obtained from Lidar processing. There were three possible reasoning for this discrepancy – firstly, the Lidar capture was incorrect including the GNSS system used, or the spatial reference used or error in post-processing; secondly, the imagery capture was incorrect including the photo centre coordinates provided or the spatial reference used; and finally, the processes used in ortho-rectification or the perspective projection of the fence-lines were the contributing factors.

Therefore, static differential GPS field survey was done over a smaller area in Morayfield to test the data. The field survey data was processed using AUSPOS and three control points found in the area were upgraded to 2nd Order Class B through the process. The evaluation of the Lidar data provided evaluation metrics that proved that the Lidar data capture was accurate for the purpose of the research.

The same GPS coordinates were and DEM derived from the high resolution Lidar was utilised for the ortho-rectification of the aerial imagery and evaluation of the results demonstrated that the issues had been resolved and the new ortho-rectification was consistent with Lidar data and GPS coordinates.

The next stage was to test the accuracy of the output of the feature extraction process. The DCDB for this area had a published accuracy value of 0.1m, which is given to DCDB upgrade done by survey accurate data. Therefore, testing of the extracted fence-line was done to compare the output lines with respect to the DCDB based on coordinate values on selected intersection points and statistical analysis.

Thus, from experimental testing of the accuracy obtained for the various processes, the results proved that the output had a high accuracy; the only limitation was the question of whether fence-lines could be considered property boundaries, but since existing DNRME process already used fence-lines to upgrade the cadastre this question was not explored.

8.5 Objective 4: Recommendations

The fourth and final objective of this research was to “*deliver recommendations on how and under what conditions remote sensing data might be employed for cadastral upgrade purposes*”.

The research project explored various airborne Lidar and Imagery data including Geiger mode Lidar data for feature extraction. The processes developed demonstrated that it is feasible to use Lidar and Imagery to extract fence-lines and that these lines can be used to upgrade the cadastral lines or polygons. Time saved in the process was evaluated, and it can definitely provide ease and accuracy for block-shift of cadastral lines.

Exploration of various data from different sources demonstrated the requirement for data that is not processed too much by the vendor and is recommended to be done by the agencies in conjunction with higher accuracy coordinates in the area of interest.

Evaluation was also done on the optimal resolution of the Lidar data for the feature extraction. There is scope for improving the developed GUI, the source code for which has been made available in IDL format.

The processes developed in this research can be recommended for use in an operational environment, provided there is Lidar data that is available. Data acquired by organisations generally tend to have multiple applications and this is one aspect that can assist with data acquisition for use in this process.

8.6 Future Research

The Lidar processes can be improved to output more fence-lines using lower resolution Lidar data. The time constraint for this research meant that all the options that could be modified in the parameter files were not explored for improving the results and default values were used throughout so there is scope for improving the results by experimenting with the default values.

The training of the support vector machine could be improved with multiple iterations of training which were not done for this research but there is potential for improvement if researched in the future.

The output of the process resulted in buildings that were extracted, which was not used in this process, but could be used to either create a standalone buildings database or to upgrade the cadastre which contained building roof-prints as part of the database.

Geiger Mode Lidar seems to be a cost effective high resolution data that could provide a wide coverage for use in the developed process. This research had access to a small image from the US, but the result showed great promise in extracting fence-lines.

The results achieved from the imagery could be improved by using Convolution Neural Network (CNN) in conjunction with SVM to filter results. A relative DEM model was used to filter the lines based on the expected range of height above ground of a fence-line, however this same process can be used for feature extraction of other features with specific height attributes.

9 References

- Aguilar, F. J., Mills, J. P., Delgado, J., Aguilar, M. A., Negreiros, J., and Pérez, J. L. (2010). Modelling vertical error in LiDAR-derived digital elevation models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1), pp 103-110.
- Aktaruzzaman, M., and Schmitt, T. G. (2010). *LIDAR-data: Automatic object detection to support urban flooding simulation*. Proceedings Canadian Geomatics Conference and Symposium of Commission I, ISPRS, Calgary, Alberta, Canada, 15-18 June, 2010.
- Axelsson, P. (1999). Processing of laser scanner data—algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2), pp 138-147.
- Axelsson, P. (2000). DEM generation from laser scanner data using adaptive TIN models. *International Archives of Photogrammetry and Remote Sensing*, 33(B4/1; PART 4), pp 111-118.
- Baligh, A., Zojj, M. V., and Mohammadzadeh, A. (2008). Bare earth extraction from airborne lidar data using different filtering methods. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(PART B3B).
- Bassier, M., Bonduel, M., Van Genechten, B., and Vergauwen, M. (2017). SEGMENTATION OF LARGE UNSTRUCTURED POINT CLOUDS USING OCTREE-BASED REGION GROWING AND CONDITIONAL RANDOM FIELDS. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-2/W8, pp 25-30. doi:10.5194/isprs-archives-XLII-2-W8-25-2017. <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W8/25/2017/>
- Biswas, R., and Sil, J. (2012). An improved canny edge detection algorithm based on type-2 fuzzy sets. *Procedia Technology*.
- Briese, C., and Pfeifer, N. (2001). *Airborne laser scanning and derivation of digital terrain models*. Proceedings of the 5th conference on optical 3D measurement techniques. 80-87.
- Brovelli, M., Cannata, M., and Longoni, U. (2002). *Managing and processing LIDAR data within GRASS*. Proceedings of the GRASS users conference.
- Brunn, A., and Weidner, U. (1998). Hierarchical Bayesian nets for building extraction using dense digital surface models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(5), pp 296-307.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp 679-698.
- Charaniya, A. P., Manduchi, R., and Lodha, S. K. (2004). *Supervised parametric classification of aerial Lidar data*. Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on. IEEE. 30-30.

- Chen, C., Li, Y., Zhao, N., Guo, J., and Liu, G. (2017). A fast and robust interpolation filter for airborne lidar point clouds. *PloS One*, 12(5), pp e0176954.
- Chen, Q., Gong, P., Baldocchi, D., and Xie, G. (2007). Filtering airborne laser scanning data with morphological methods. *Photogrammetric Engineering & Remote Sensing*, 73(2), pp 175-185.
- Cheng, M., and Weng, Q. (2017). Urban Road Extraction from Combined Data Sets of High-Resolution Satellite Imagery and Lidar Data Using GEOBIA. *Integrating Scale in Remote Sensing and GIS*, pp 283.
- Ding, L., and Goshtasby, A. (2001). On the Canny edge detector. *Pattern Recognition*, 34(3), pp 721-725.
- Du, S., Zhang, Y., Qin, R., Yang, Z., Zou, Z., Tang, Y., and Fan, C. (2016). Building Change Detection Using Old Aerial Images and New LiDAR Data. *Remote Sensing*, 8(12), pp 1030.
- Elmqvist, M. (2002). Ground surface estimation from airborne laser scanner data using active shape models. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A), pp 114-118.
- Elmqvist, M., Jungert, E., Lantz, F., Persson, A., and Soderman, U. (2001). Terrain modelling and analysis using laser scanner data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W4), pp 219-226.
- Fernandez, J., Singhanian, A., Caceres, J., Slatton, K., Starek, M., and Kumar, R. (2007). An overview of lidar point cloud processing software. *GEM Center Report No. Rep_2007-12-001, University of Florida*.
- Flood, M. (2004). ASPRS Guidelines: Vertical Accuracy Reporting for LiDAR Data. *American Society for Photogrammetry and Remote Sensing*, 1.
- Gerke, M., and Xiao, J. (2014). Fusion of airborne laserscanning point clouds and images for supervised and unsupervised scene classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, pp 78-92.
- Ghamisi, P., and Höfle, B. (2017). LiDAR Data Classification Using Extinction Profiles and a Composite Kernel Support Vector Machine. *IEEE Geoscience and Remote Sensing Letters*.
- Green, B. (2002). Canny Edge Detection Tutorial. Retrieved from
- Haugerud, R. A., and Harding, D. (2001). Some algorithms for virtual deforestation (VDF) of LIDAR topographic survey data. *International Archives of Photogrammetry and Remote Sensing*, 34(3/W4), pp 211-218.
- Herмосilla, T., Ruiz, L. A., Recio, J. A., and Estornell, J. (2011). Evaluation of automatic building detection approaches combining high resolution images and LiDAR data. *Remote Sensing*, 3(6), pp 1188-1210.

- Hu, X., Tao, C. V., and Hu, Y. (2004). Automatic road extraction from dense urban area by integrated processing of high resolution imagery and lidar data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35.
- Hu, Y. (2003). *Automated extraction of digital terrain models, roads and buildings using airborne LiDAR data*. (PhD Thesis), University of Calgary.
- Huang, X., Zhang, L., and Gong, W. (2011). Information fusion of aerial images and LIDAR data in urban areas: vector-stacking, re-classification and post-processing approaches. *International Journal of Remote Sensing*, 32(1), pp 69-84.
- Juneja, M., and Sandhu, P. S. (2009). Performance evaluation of edge detection techniques for images in spatial domain. *International Journal of Computer Theory and Engineering*, 1(5), pp 1793-8201.
- Kim, E., and Medioni, G. (2011). Urban scene understanding from aerial and ground LIDAR data. *Machine Vision and Applications*, 22(4), pp 691-703.
- Kraus, K., and Pfeifer, N. (1998). Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(4), pp 193-203.
- Kraus, K., and Pfeifer, N. (2001). Advanced DTM generation from LIDAR data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(3/W4), pp 23-30.
- Li, N., Liu, C., Pfeifer, N., Yin, J., Liao, Z., and Zhou, Y. (2016). Tensor Modeling Based Airborne Lidar Data Classification. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41.
- Liu, X. (2008). Airborne LiDAR for DEM generation: some critical issues. *Progress in Physical Geography*, pp 31-49.
- Liu, X., Zhang, Z., Peterson, J., and Chandra, S. (2007). *The effect of LiDAR data density on DEM accuracy*. Proceedings of the International Congress on Modelling and Simulation (MODSIM07), Christchurch, New Zealand.
- Marr, D., and Hildreth, E. (1980). Theory of Edge Detection. *Proceedings of the Royal Society of London, B 207*, pp 187-217.
- Meng, X., Currit, N., Wang, L., and Yang, X. (2010). *Object-oriented residential building land-use mapping using lidar and aerial photographs*. American Society of Photogrammetry and Remote Sensing 2010 Annual Conference, San Diego. 26-30.
- Meng, X., Currit, N., and Zhao, K. (2010). Ground filtering algorithms for airborne LiDAR data: A review of critical issues. *Remote Sensing*, 2(3), pp 833-860.
- Montealegre, A. L., Lamelas, M. T., and de la Riva, J. (2015). A comparison of open-source LiDAR filtering algorithms in a mediterranean forest environment. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(8), pp 4072-4085.

- Peli, T., and Malah, D. (1982). A study of edge detection algorithms. *Computer Graphics and Image Processing*, 20, pp 1-21.
- Peng, D., and Zhang, Y. (2016). Building Change Detection by Combining Lidar Data and Ortho Image. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41.
- Pfeifer, N., Reiter, T., Briese, C., and Rieger, W. (1999). Interpolation of high quality ground models from laser scanner data in forested areas. *International Archives of Photogrammetry and Remote Sensing*, 32(3/W14), pp 31-36.
- Pfeifer, N., Stadler, P., and Briese, C. (2001). *Derivation of digital terrain models in the SCOP++ environment*. Proceedings of OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Terrain Models, Stockholm, Sweden.
- Robinson, N., Regetz, J., and Guralnick, R. P. (2014). EarthEnv-DEM90: A nearly-global, void-free, multi-scale smoothed, 90m digital elevation model from fused ASTER and SRTM data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, pp 57-67.
- Roggero, M. (2001). Airborne laser scanning-clustering in raw data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W4), pp 227-232.
- Rottensteiner, F., Trinder, J., Clode, S., and Kubik, K. (2003). *Detecting buildings and roof segments by combining LIDAR data and multispectral images*. Image and Vision Computing New Zealand 2003.
- Schenk, T., and Csathó, B. (2002). Fusion of LIDAR data and aerial imagery for a more complete surface description. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(3/A), pp 310-317.
- Sharifi, M., Fathy, M., and Mahmoudi, M. T. (2002). *A classified and comparative study of edge detection algorithms*. Proceedings of the International Conference on Information Technology: Coding and Computing. 117-120.
- Shrivakshan, G. T., and Chandrasekar, C. (2012). A comparison of various edge detection techniques used in image processing. *IJCSI International Journal of Computer Science Issues*, 9(5), pp 269-276.
- Silván-Cardenás, J. L., and Wang, L. (2006). A multi-resolution approach for filtering LiDAR altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(1), pp 11-22.
- Sithole, G., and Vosselman, G. (2001). Filtering of laser altimetry data using a slope adaptive filter. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W4), pp 203-210.
- Sithole, G., and Vosselman, G. (2003a). *Automatic structure detection in a point-cloud of an urban landscape*. Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop. IEEE. 67-71.

- Sithole, G., and Vosselman, G. (2003b). Report: ISPRS comparison of filters. *ISPRS Commission III, Working Group, 3*.
- Sohn, G., and Dowman, I. (2002). Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(3/A)*, pp 336-344.
- Sohn, G., and Dowman, I. (2007). Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction. *ISPRS Journal of Photogrammetry and Remote Sensing, 62(1)*, pp 43-63.
- Stoker, J. M., Abdullah, Q. A., Nayegandhi, A., and Winehouse, J. (2016). Evaluation of single photon and Geiger mode Lidar for the 3D Elevation Program. *Remote Sensing, 8(9)*, pp 767.
- Tao, C., and Hu, Y. (2001). *A review of post-processing algorithms for airborne LIDAR data*. CD-ROM Proceedings of ASPRS Annual Conference, April 23-27, 2001, St. Louis Missouri.
- Tarsha-Kurdi, F., Landes, T., and Grussenmeyer, P. (2007). *Hough-transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data*. Proceedings of the ISPRS Workshop on Laser Scanning. 407-412.
- Ullrich, A., and Pfennigbauer, M. (2016). *Linear LIDAR versus Geiger-mode LIDAR: impact on data properties and data quality*. SPIE Defense+ Security. International Society for Optics and Photonics. 983204-983204-983217.
- Vosselman, G. (2000). Slope based filtering of laser altimetry data. *International Archives of Photogrammetry and Remote Sensing, 33(B3)*, pp 935-942.
- Wack, R., and Wimmer, A. (2002). Digital Terrain Models from Airborne Laser Scanner Data- A Grid Based Approach. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(3/B)*, pp 293-296.
- Wang, H., and Glennie, C. (2015). Fusion of waveform LiDAR data and hyperspectral imagery for land cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing, 108(Supplement C)*, pp 1-11. doi:<https://doi.org/10.1016/j.isprsjprs.2015.05.012>.
<http://www.sciencedirect.com/science/article/pii/S0924271615001495>
- Wang, L., and Neumann, U. (2009). *A robust approach for automatic registration of aerial images with untextured aerial lidar data*. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference. IEEE. 2623-2630.
- Wang, O., Lodha, S. K., and Helmbold, D. P. (2006). *A Bayesian Approach to Building Footprint Extraction from Aerial Lidar Data*. 3D Data Processing, Visualization, and Transmission, Third International Symposium on. IEEE. 192-199.
- Zhang, K., and Whitman, D. (2005). Comparison of three algorithms for filtering airborne lidar data. *Photogrammetric Engineering & Remote Sensing, 71(3)*, pp 313-324.

- Zhang, K., Whitman, D., Shyu, M.-L., Yan, J., and Zhang, C. (2003). A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4), pp pp 872-882.
- Zhou, G., and Zhou, X. (2014). Seamless fusion of LiDAR and aerial imagery for building extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11), pp 7393-7407.
- Ziou, D., and Tabbone, S. (1998). Edge detection techniques-an overview. *Pattern Recognition and Image Analysis*, 8, pp 537-559.

10.2 Camera Calibration Report Extract

AICP65+PRO_201411209_12239531

Camera Calibration

Block Adjustment - Version 8.0.3.51776 (64bit), build #73 of 2017-01-18 03:42 at 20.03.2018 13:20

Camera: AICP65+PRO_201411209_1223_calibration_13:19:56_20/03/2018

Manufacturer: TAC Px CaptureOne v.x

Ser.No.: 201411209

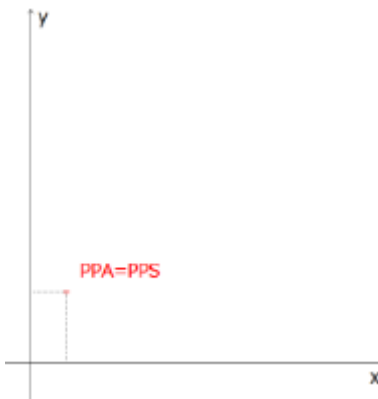
sensor width: 8984

sensor height: 6732

pixel size: 6.00 x 6.00 [μm]

Internal Geometry

	original	calibrated	Std.Dev.
Focal Length:	51.4798 mm	51.4921 mm	+/- 2.0260 μm
Princ. Point x:	0.2754 mm	0.2731 mm	+/- 1.6828 μm
Princ. Point y:	0.5494 mm	0.5436 mm	+/- 1.5122 μm



Additional Parameters

Physical

correction for radial distortion

$$DX = X * (K_0 + K_1 * R^2 + K_2 * R^4 + K_3 * R^6 + K_4 * R^8 + K_5 * R^{10} + K_6 * R^{12} + K_7 * R^{14})$$

$$DY = Y * (K_0 + K_1 * R^2 + K_2 * R^4 + K_3 * R^6 + K_4 * R^8 + K_5 * R^{10} + K_6 * R^{12} + K_7 * R^{14})$$

correction for decentering distortion

$$DX = P_1 * (R^2 + 2 * X^2) + 2 * P_2 * X * Y$$

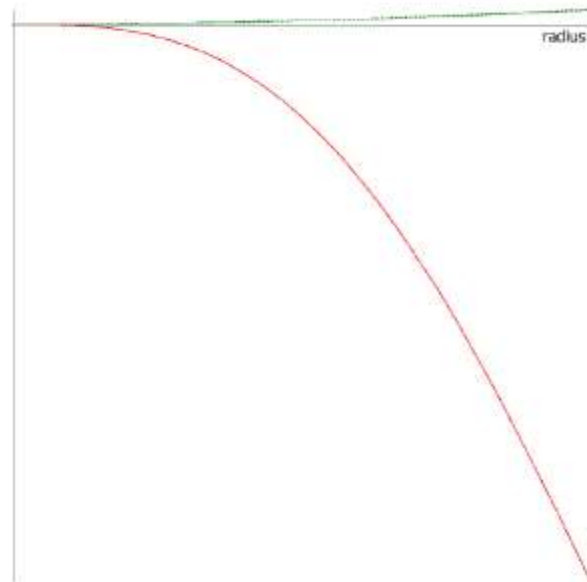
$$DY = 2 * P_1 * X * Y + P_2 * (R^2 + 2 * Y^2)$$

with $R = (X * X + Y * Y)^{1/2}$

X and Y are with respect to principal point (PPA=PPS)

Parameter	Value	Std.Dev.
K₀	0.00000e+00	
K₁	-1.46895e-05	+/- 9.3e-08
K₂	4.21514e-09	+/- 1.9e-10
K₃	-2.19688e-13	+/- 1.1e-13
P₁	-8.36091e-06	+/- 2.0e-07
P₂	-1.84371e-06	+/- 1.8e-07

Radius [mm]	Distortion [um]
0.0000	-0.0000
2.0000	-0.1174
4.0000	-0.9358
6.0000	-3.1402
8.0000	-7.3834
10.0000	-14.2702
12.0000	-24.3425
14.0000	-38.0641
16.0000	-55.8073
18.0000	-77.8389
20.0000	-104.3088
22.0000	-135.2385
24.0000	-170.5117
26.0000	-209.8655
28.0000	-252.8840
30.0000	-298.9931
32.0000	-347.4573
34.0000	-397.3781



Distortion Error of radial symmetric components above parameters

The green line show the magnitude of the decentring distortions on the four image diagonals. This gives an impression of what will be missed if only radial distortion components are used.

10.3 Photos Lidar + Imagery Data Capture





10.4 Photos GPS Field Survey



10.5 Processing in Lastools

Convert LAS (.las) files to Zipped LAS (.laz) files for ease of processing: The zipped .laz files are about 6 times smaller in size.

```
lastile -i Input_Folder\*.las -odir Out_Folder\Laz -olaz
```

Compute information and check density of point cloud of the provided tiles

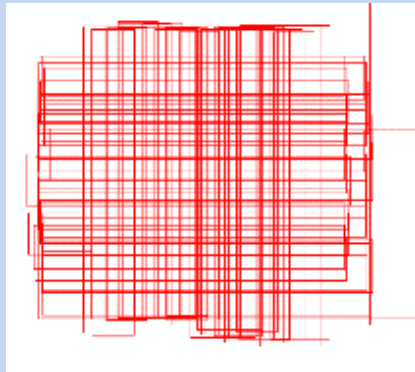
```
lasinfo -i Input_Folder\*.laz -cd -otxt -odir Out_Folder\Laz -odix_info -cores 7
```

Create regular tiles from original and place them in raw folder: An overlap buffer is specified for edge-matching. Flight line information is maintained. As the free version of LASTools does not allow to work with a point cloud greater than 1Million points, so it is sometimes necessary to create smaller tiles (250mx250m in this case).

```
lastile -i Input_Folder\*.laz -files_are_flightlines -tile_size 250 -buffer 50 -odir Out_Folder\tiles_raw\ -olaz
```

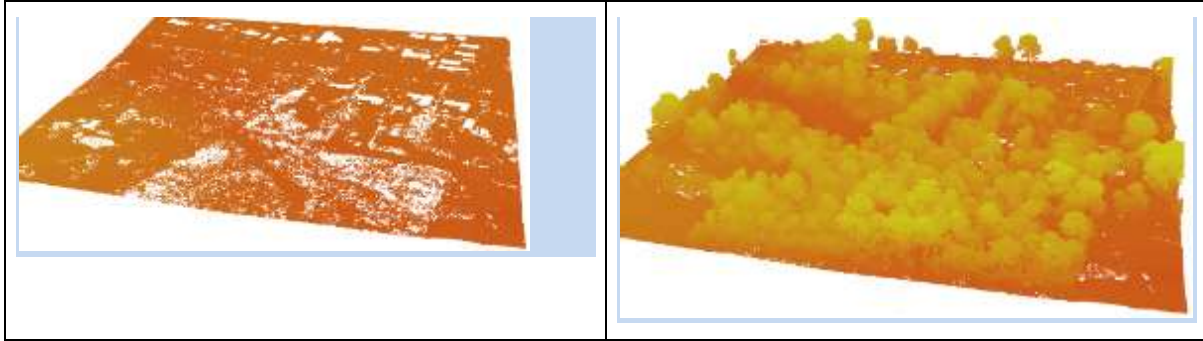
Create individual shapefiles of the polygon boundary of Las files: Run ArcGIS model to include filename as an attribute; Run Merge to create a single shapefile, delete the temporary individual files

```
lasboundary -i Input_Folder\tiles_raw\*.laz -odir Out_Folder\temp -use_tile_bb -oshp
```



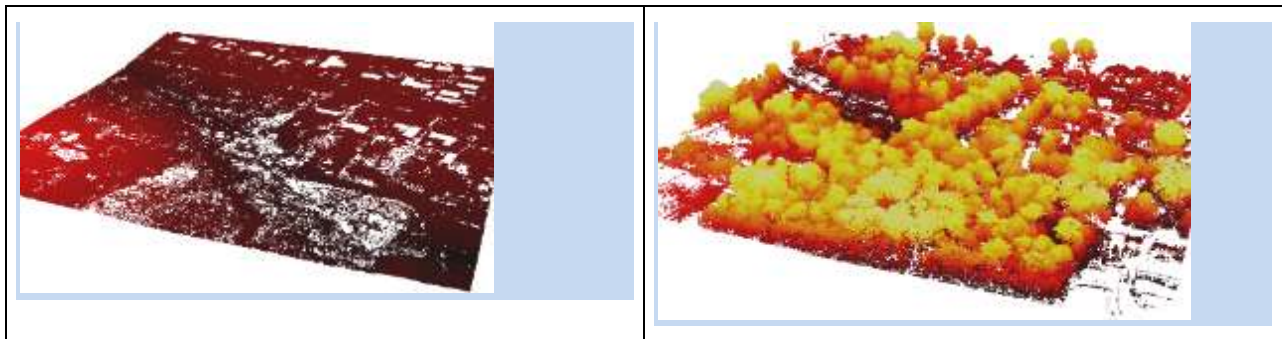
Extract bare earth: By classifying ground points in class=2 and non-ground points in class=1. The cores 7 is for multi-thread processing but depends on the number of cores available (e.g. this computer has 8 cores so 7 are used for parallel processing)

```
lasground -i Input_Folder\tiles_raw\*.laz -town -coarse -odir Out_Folder\Laz\tiles_ground -olaz -cores 7
```



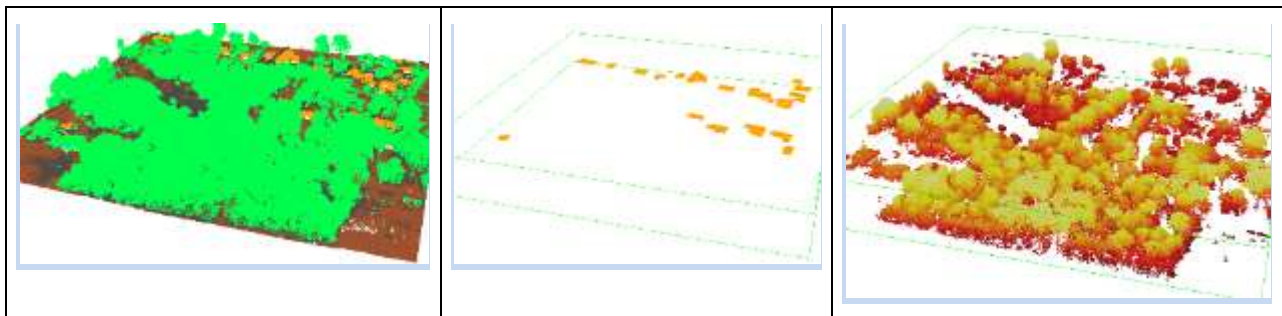
Compute the height of each LAS point above the ground from ground classified points to identify and construct ground TIN:

```
lasheight -i Input_Folder \Laz\tiles_ground\*.laz -drop_below -2 -drop_above 30 -odir Out_Folder\Laz\tiles_height -olaz -cores 7
```



Classify buildings and vegetation: From points whose bare earth computation has already been completed

```
lasclassify -i Input_Folder\Laz\tiles_height\*.laz -step 3 -odir Out_Folder\Laz\tiles_classified -olaz -cores 7
```



Classified – Ground, Building, Vegetation

Create non-overlapping square tiles

```
lastile -i Input_Folder\tiles_classified\*.laz -set_user_data 0 -remove_buffer -odir Out_Folder\Laz\tiles_final -olaz
```

Compute information and check density of point cloud of the new tiles

```
lasinfo -i Input_Folder\tiles_final\*.laz -cd -otxt -odir Out_Folder\Laz\tiles_final -odix_info -cores 7
```

Create individual shapefiles of the polygon boundary of the new tiles: Run ArcGIS model to include filename as an attribute; Run Merge to create a single shapefile, delete the temporary individual files

```
lasboundary -i Input_Folder\Laz\tiles_final\*.laz -odir Out_Folder\Laz\temp -use_tile_bb -oshp
```

Create DTM and DSM: Read LIDAR points from the LAS/LAZ format, triangulate them temporarily into a TIN, and rasterise the TIN onto a DEM (Both DTM and DSM). This process creates a Digital Terrain Model (DTM) of the ground surface

```
las2dem -i Input_Folder\Laz\tiles_classified\*.laz -keep_class 2 -thin_with_grid 0.1 -extra_pass -odir Out_Folder\Laz\tiles_dtm -obil -cores 7
```

This process creates a Digital Surface Model (DSM) of the non-ground surface

```
las2dem -i Input_Folder\Laz\tiles_classified\*.laz -first_only -thin_with_grid 0.1 -extra_pass -use_tile_bb -odir Out_Folder\tiles_dsm -obil -cores 7
```

Create a combined raster of the DTMs created earlier: If the number of points are above one million then black diagonal lines are introduced, so it might be desirable to create individual DTMs and merge them later.

```
lasgrid -i Input_Folder\Laz\tiles_dtm\*.bil -merged -odir Out_Folder\Laz\tiles_dtm -o dtm.tif
```

Create a combined raster of the DSMs created earlier: If the number of points are above 1Million then black diagonal lines are introduced, so it might be desirable to create individual DSMs and merge them later.

```
lasgrid -i Input_Folder\Laz\tiles_dsm\*.bil -merged -odir Out_Folder\Laz\tiles_dsm -o dsm.tif
```

Create a combined Hillshade of the DTMs created earlier: If the number of points are above 1Million then black diagonal lines are introduced, so it might be desirable to create individual hillshade files and merge them later.

```
blast2dem -i Input_Folder\Laz\tiles_dtm\*.bil -merged -hillshade -odir Out_Folder\tiles_dtm -o dtm_hillshade_raster.png
```

Create a combined Hillshade of the DSMs created earlier: If the number of points are above 1Million then black diagonal lines are introduced, so it might be desirable to create individual Hillshade files and merge them later.

```
blast2dem -i Input_Folder\Laz\tiles_dsm\*.bil -merged -hillshade -odir Out_Folder\tiles_dsm -o dsm_hillshade_raster.png
```



Create a combined contour of the DTM at 1m contour intervals: Lines smaller than 5 units and area smaller than 1 unit are cleared. Lengths smaller than 0.5 units are simplified.

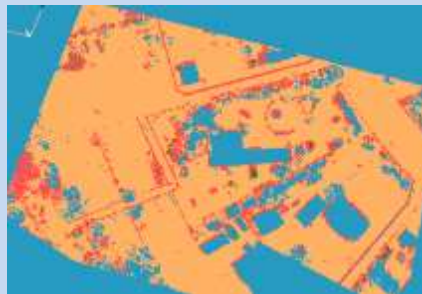
```
blast2iso -i Input_Folder\Laz\tiles_dtm\*.bil -merged -iso_every 1 -simplify_length 0.5 -simplify_area 1 -clean 5 -odir Out_Folder\tiles_dtm -o dtm_contours_raster_1m.shp
```

Create a combined contour of the DSM at 3m contour intervals: Lines smaller than 5 units and area smaller than 1 unit are cleared. Lengths smaller than 0.5 units are simplified.

```
blast2iso -i Input_Folder\Laz\tiles_dsm\*.bil -merged -iso_every 3 -simplify_length 0.5 -simplify_area 1 -clean 5 -odir Out_Folder\tiles_dsm -o dsm_contours_raster_3m.shp
```

Filter points above ground level between $0.5 < \text{elevation} < 2.0$: This filters all points outside of 0.5-2m AGL for further processing.

```
lasheight -i Input_Folder\Laz\temp\*.laz -drop_below 0.5 -drop_above 2 -odir Out_Folder\temp\ht -olaz -cores 7
```



10.6 Input Parameters for the Developed GUI

Three .json files have been created for the input parameters that are read by the graphical user interface (GUI). These parameters are applicable to both ENVI and non-ENVI version. An additional .json file exists for imagery based application.

JSON File Contents: High Density Data Parameters

```
{
  "FENCES": {
    "FENCES_MAXIMUM_HEIGHT": 2.5000000,
    "FENCES_MINIMUM_HEIGHT": 0.6000000,
    "FENCES_MINIMUM_NUM_POINTS": 20,
    "FENCES_PERPENDICULAR_TOLERANCE": 0.3000000,
    "FENCES_MAXIMUM_GAP": 0.7500000,
    "FENCES_CLUSTER_DISTANCE": 0.6000000,
    "FENCES_MINIMUM_LENGTH": 4.0000000,
    "FENCES_VEG_REMOVAL_KERNEL_RADIUS": 2.0000000,
    "FENCES_VEG_REMOVAL_GRID": 5,
    "FENCES_VEG_REMOVAL_MAXIMUM_COVERAGE": 75,
    "SVM_MODEL": "RBF-Dense-50.model"
  },
  "BUILDINGS": {
    "BUILDINGS_MINIMUM_AREA": 10.000000,
    "BUILDINGS_NEAR_GROUND_FILTER_WIDTH": 300.00000,
    "BUILDINGS_PLANE_SURFACE_TOLERANCE": 30.000000,
    "BUILDINGS_POINTS_IN_RANGE": 0.000000
  },
  "DEM": {
    "DEM_CONSTANT_HEIGHT_OFFSET": 2.0000000,
    "DEM_FILTER_DATABASE_EDGES": 0,
    "DEM_FILTER_LOWER_POINTS": 10,
    "DEM_NEAR_TERRAIN_CLASSIFICATION": 50.000000,
    "DEM_SENSITIVITY": 30.000000,
    "DEM_VARIABLE_SENSITIVITY_ALGORITHM": 0
  },
  "DSM": {
    "DSM_USE_POWERLINES_POINTS": 1,
    "DSM_GENERATE": 0
  },
  "POWERLINES": {
    "POWERLINES_FILTER_BY_MIN_JOINED_LENGTH": 0,
    "POWERLINES_MIN_JOINED_LENGTH": 20.000000,
    "POWERLINES_FILTER_TREES_BELOW_POWERLINES": 1,
    "POWERLINES_MIN_LENGTH": 10.000000,
    "POWERLINES_SEARCH_LOWKV": 0,
    "POWERLINES_SEARCH_WIDE": 0,
    "POWERPOLES_CLASS_EXTEND_TOP": 0.0000000,
    "POWERPOLES_CLASS_RADIUS_LOW": 3.0000000,
    "POWERPOLES_EXTEND_WIRES_DISTANCE": 40.000000,
    "POWERPOLES_MAX_RADIUS_TOP": 10.000000,
    "POWERPOLES_SEARCH_ADDITIONAL": 0
  },
  "TREES": {
    "TREES_MAX_HEIGHT": 5000.0000,
    "TREES_MIN_HEIGHT": 130.00000,
    "TREES_MAX_RADIUS": 600.00000,
    "TREES_MIN_RADIUS": 200.00000
  }
}
```

JSON File Contents: Medium Density parameters

```
{
  "FENCES": {
    "FENCES_MAXIMUM_HEIGHT": 2.5000000,
    "FENCES_MINIMUM_HEIGHT": 0.6000000,
    "FENCES_MINIMUM_NUM_POINTS": 15,
    "FENCES_PERPENDICULAR_TOLERANCE": 0.3000000,
    "FENCES_MAXIMUM_GAP": 1.7500000,
    "FENCES_CLUSTER_DISTANCE": 1.2000000,
    "FENCES_MINIMUM_LENGTH": 3.0000000,
    "FENCES_VEG_REMOVAL_KERNEL_RADIUS": 2.0000000,
    "FENCES_VEG_REMOVAL_GRID": 5,
    "FENCES_VEG_REMOVAL_MAXIMUM_COVERAGE": 75,
    "SVM_MODEL": "RBF-Dense-50.model"
  },
  "BUILDINGS": {
    "BUILDINGS_MINIMUM_AREA": 10.000000,
    "BUILDINGS_NEAR_GROUND_FILTER_WIDTH": 300.00000,
    "BUILDINGS_PLANE_SURFACE_TOLERANCE": 30.000000,
    "BUILDINGS_POINTS_IN_RANGE": 0.0000000
  },
  "DEM": {
    "DEM_CONSTANT_HEIGHT_OFFSET": 2.0000000,
    "DEM_FILTER_DATABASE_EDGES": 0,
    "DEM_FILTER_LOWER_POINTS": 10,
    "DEM_NEAR_TERRAIN_CLASSIFICATION": 50.000000,
    "DEM_SENSITIVITY": 30.000000,
    "DEM_VARIABLE_SENSITIVITY_ALGORITHM": 0
  },
  "DSM": {
    "DSM_USE_POWERLINES_POINTS": 1,
    "DSM_GENERATE": 0
  },
  "POWERLINES": {
    "POWERLINES_FILTER_BY_MIN_JOINED_LENGTH": 0,
    "POWERLINES_MIN_JOINED_LENGTH": 20.000000,
    "POWERLINES_FILTER_TREES_BELOW_POWERLINES": 1,
    "POWERLINES_MIN_LENGTH": 10.000000,
    "POWERLINES_SEARCH_LOWKV": 0,
    "POWERLINES_SEARCH_WIDE": 0,
    "POWERPOLES_CLASS_EXTEND_TOP": 0.0000000,
    "POWERPOLES_CLASS_RADIUS_LOW": 3.0000000,
    "POWERPOLES_EXTEND_WIRES_DISTANCE": 40.000000,
    "POWERPOLES_MAX_RADIUS_TOP": 10.000000,
    "POWERPOLES_SEARCH_ADDITIONAL": 0
  },
  "TREES": {
    "TREES_MAX_HEIGHT": 5000.0000,
    "TREES_MIN_HEIGHT": 130.00000,
    "TREES_MAX_RADIUS": 600.00000,
    "TREES_MIN_RADIUS": 200.00000
  }
}
```

JSON File Contents: Sparse Density Parameters:

```
{
  "PREPROCESS": {
    "HILLSHADE_GSD": 0.2,
    "HILLSHADE_MAX_HEIGHT": 50.0,
    "HILLSHADE_MIN_HEIGHT": -0.1
  },
  "FENCES": {
```

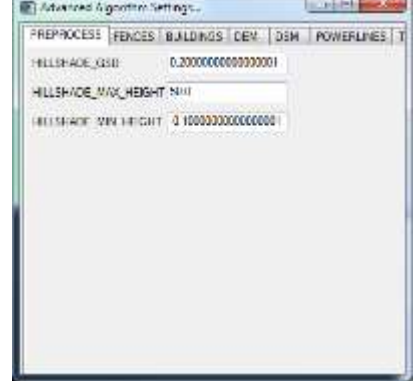



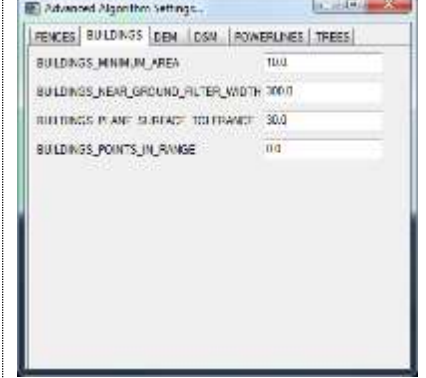
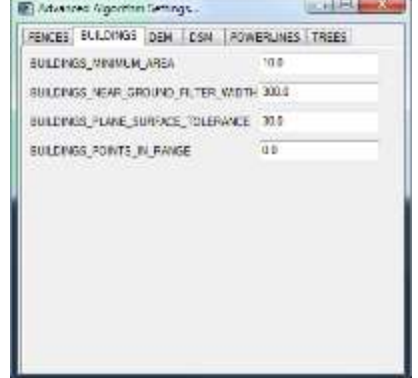
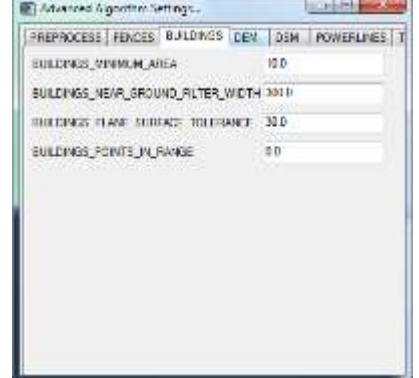
```

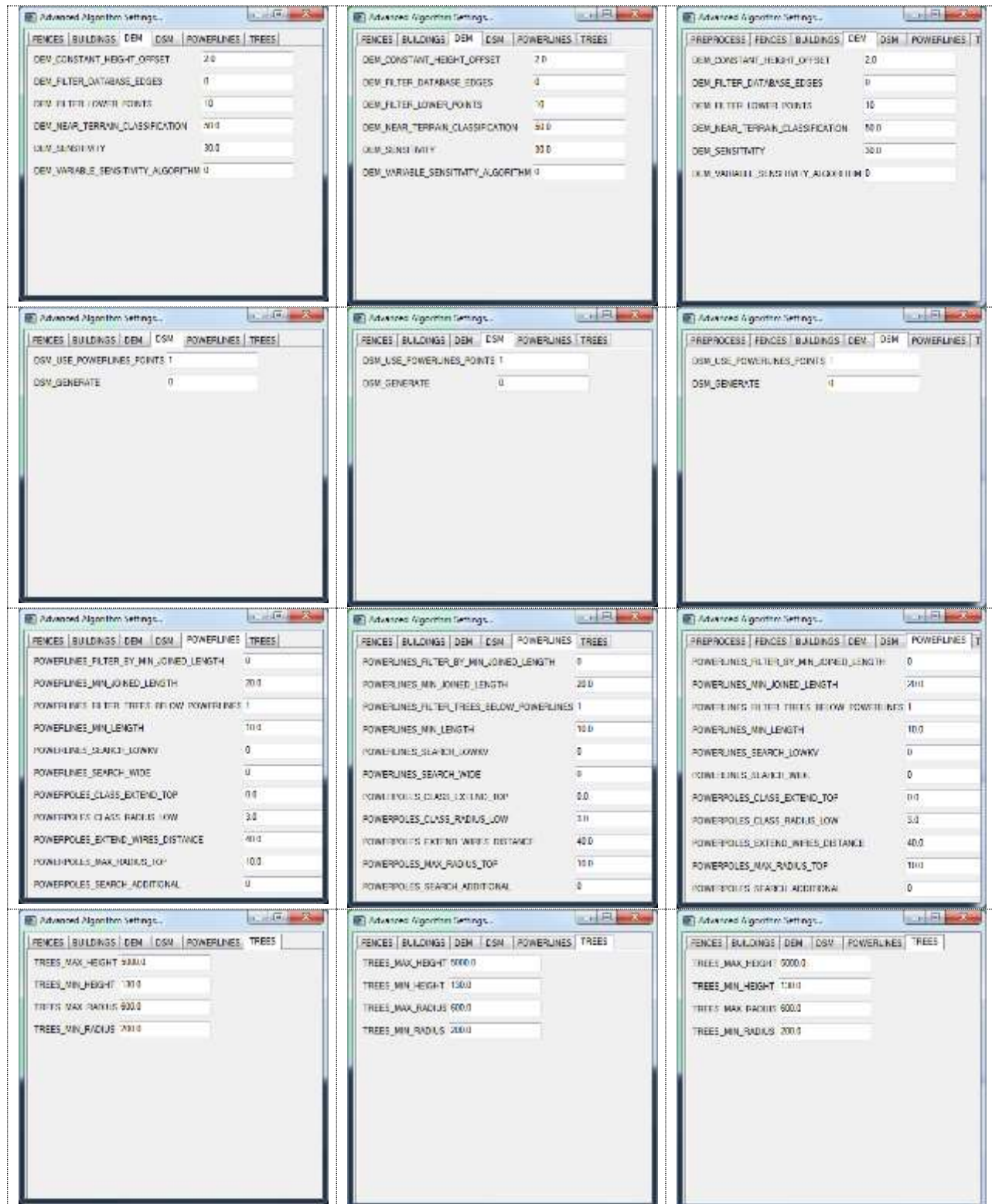
    "FENCES_MAXIMUM_HEIGHT": 2.5000000,
    "FENCES_MINIMUM_HEIGHT": 0.6000000,
    "FENCES_MINIMUM_NUM_POINTS": 10,
    "FENCES_PERPENDICULAR_TOLERANCE": 0.5000000,
    "FENCES_MAXIMUM_GAP": 6.0000000,
    "FENCES_CLUSTER_DISTANCE": 5.0000000,
    "FENCES_MINIMUM_LENGTH": 8.0000000,
    "FENCES_VEG_REMOVAL_KERNEL_RADIUS": 6.0000000,
    "FENCES_VEG_REMOVAL_GRID": 5,
    "FENCES_VEG_REMOVAL_MAXIMUM_COVERAGE": 60,
    "SVM_MODEL": "RBF-Dense-50.model"
  },
  "BUILDINGS": {
    "BUILDINGS_MINIMUM_AREA": 10.000000,
    "BUILDINGS_NEAR_GROUND_FILTER_WIDTH": 300.00000,
    "BUILDINGS_PLANE_SURFACE_TOLERANCE": 30.000000,
    "BUILDINGS_POINTS_IN_RANGE": 0.000000
  },
  "DEM": {
    "DEM_CONSTANT_HEIGHT_OFFSET": 2.0000000,
    "DEM_FILTER_DATABASE_EDGES": 0,
    "DEM_FILTER_LOWER_POINTS": 10,
    "DEM_NEAR_TERRAIN_CLASSIFICATION": 50.000000,
    "DEM_SENSITIVITY": 30.000000,
    "DEM_VARIABLE_SENSITIVITY_ALGORITHM": 0
  },
  "DSM": {
    "DSM_USE_POWERLINES_POINTS": 1,
    "DSM_GENERATE": 0
  },
  "POWERLINES": {
    "POWERLINES_FILTER_BY_MIN_JOINED_LENGTH": 0,
    "POWERLINES_MIN_JOINED_LENGTH": 20.000000,
    "POWERLINES_FILTER_TREES_BELOW_POWERLINES": 1,
    "POWERLINES_MIN_LENGTH": 10.000000,
    "POWERLINES_SEARCH_LOWKV": 0,
    "POWERLINES_SEARCH_WIDE": 0,
    "POWERPOLES_CLASS_EXTEND_TOP": 0.0000000,
    "POWERPOLES_CLASS_RADIUS_LOW": 3.0000000,
    "POWERPOLES_EXTEND_WIRES_DISTANCE": 40.000000,
    "POWERPOLES_MAX_RADIUS_TOP": 10.000000,
    "POWERPOLES_SEARCH_ADDITIONAL": 0
  },
  "TREES": {
    "TREES_MAX_HEIGHT": 5000.0000,
    "TREES_MIN_HEIGHT": 130.00000,
    "TREES_MAX_RADIUS": 600.00000,
    "TREES_MIN_RADIUS": 200.00000
  }
}

```


Comparison of the parameters

These parameters are read in the GUI and a comparison of all the parameters together are shown below. If any values need to be changed it is best to change it in the .json file.

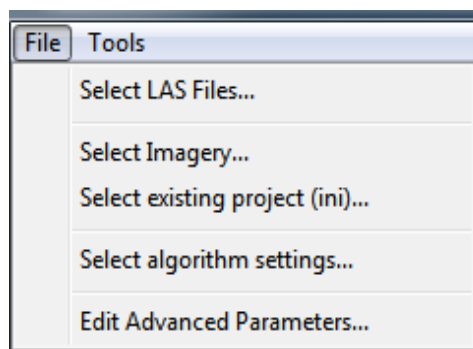
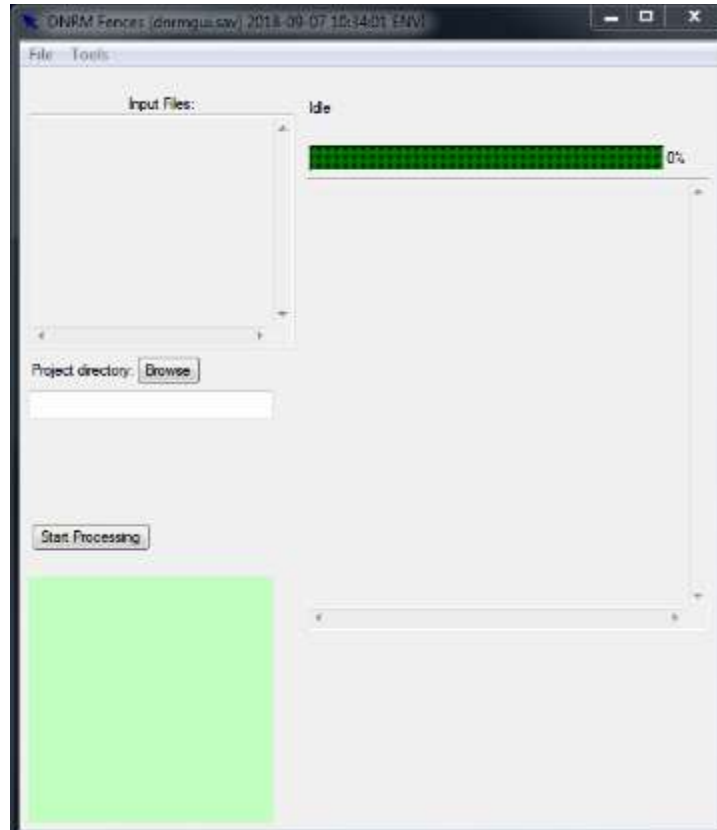
High Density	Medium	Low
		
		
		



10.7 How to Process Lidar data in the Developed GUI

Step 1: As this is an IDL based coding, ensure that IDL is installed. With an ENVI installation, IDL is installed along-with ENVI specific libraries, but the software works with the open source version too, but has slightly different steps, which will be made clear here:

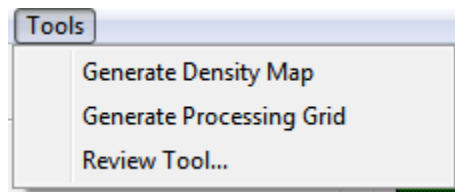
Step 2: Open dnrmgui.sav; the following window opens, and click File menu to start



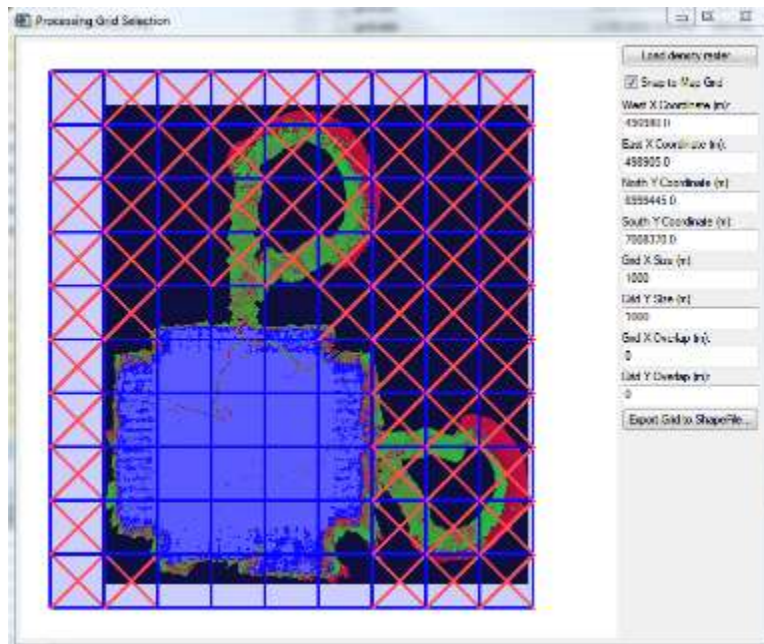
Step 3: For Lidar processing press “Select Las files”; for non-envi Lidar processing provide preclassified Lidar; and for imagery processing “Select Imagery” which will ask for a relative DEM file along-with the imagery;

Step 4: Select algorithm settings which is the .json files based on the density of the Lidar. For an imagery based processing, select imagery-relevant .json files supplied in the same folder as the software;

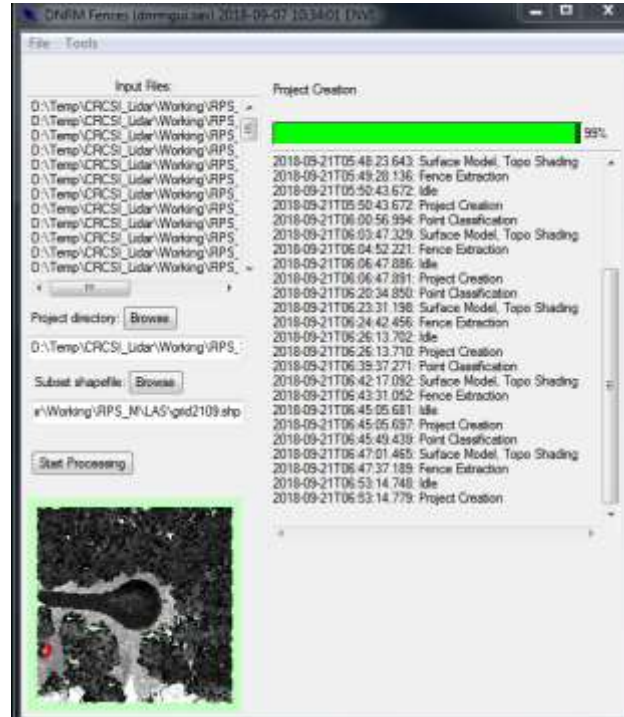
Step 5: Next open the Tools menu and select “Generate Density Map” (if processing Lidar). For Imagery or non-ENVI based implementation skip this step:



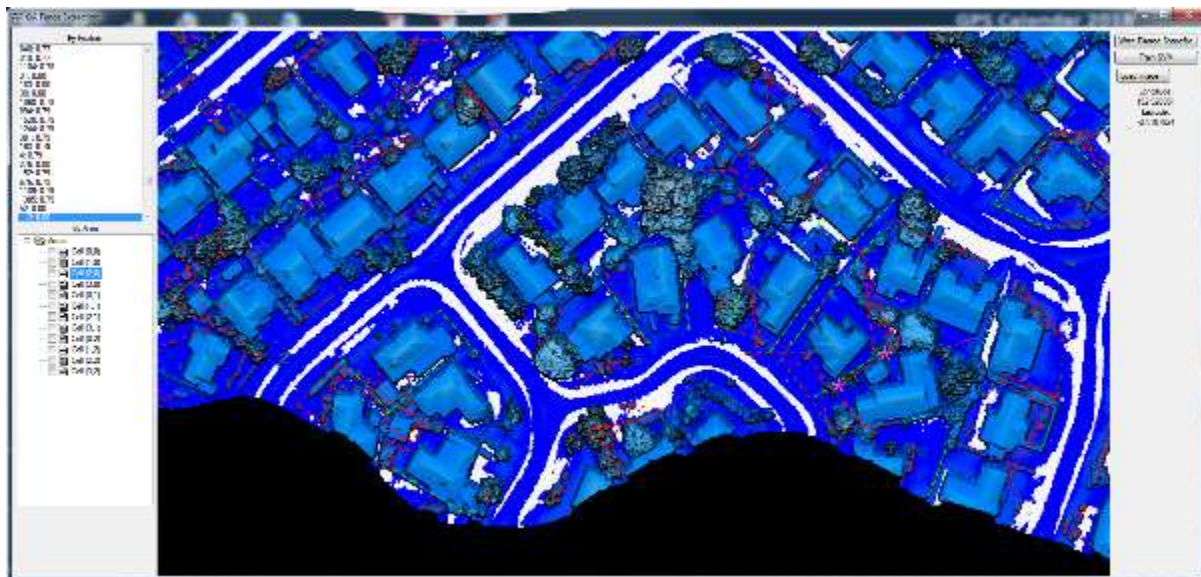
Step 6: Next select “Generate Processing Grid” from the Tools menu. This provides an option to select the area to be processed and it is exported as a shapefile. For non-ENVI based implementation, a shapefile created externally should be provided to define the area of interest. Exit the processing grid selection window to return to the Main window.



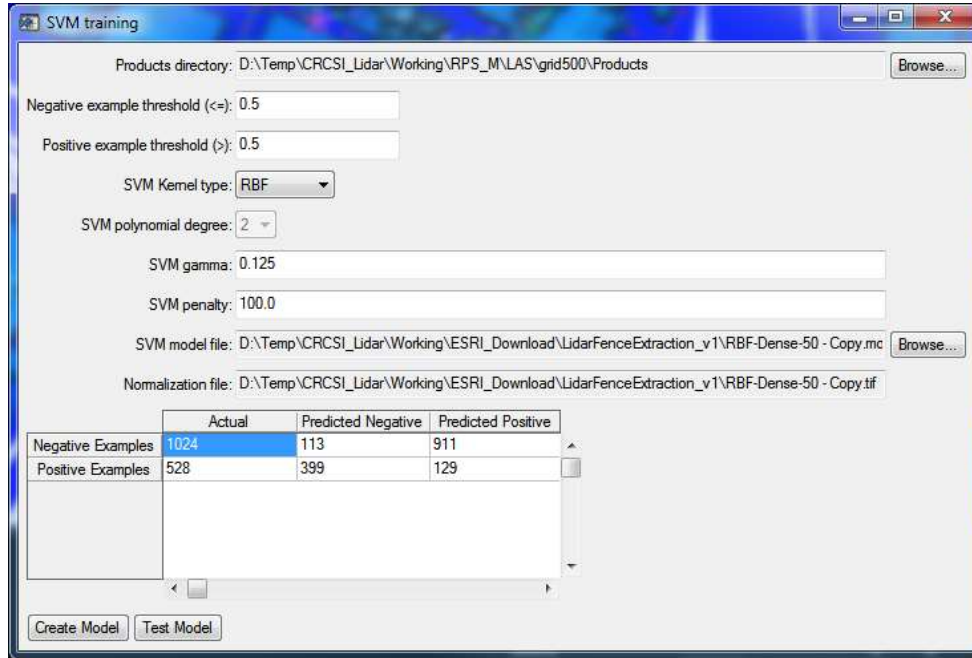
Step 7: Select “Start Processing” button; the display window shows the steps being performed;



Step 7: Select “Review Tool” from the Tools menu for quality assurance of the data as well as training the support vector machine (SVM).



Step 8: After enough training samples have been selected, export the SVM training model; and export the shapefile whose probability values have been manually edited during the training.



Check the Log files for details of the implementation if required, this file is created at the location of the Lidar data and the name starts with the date and time of the process, so it should be easy to track if there are a lot of them.

```

log_20181017T122254_056.txt
1 2018-10-17T12:23:02.332: Wrote subset LAS file:
2 2018-10-17T12:23:02.332: D:\Temp\CRCST_Lidar\Working\RPS\Calibration site data\LAS\grid_17_100\cld_170720_01_035547_001_sub.las
3 2018-10-17T12:23:11.926: Wrote subset LAS file:
4 2018-10-17T12:23:11.926: D:\Temp\CRCST_Lidar\Working\RPS\Calibration site data\LAS\grid_17_100\cld_170720_01_035626_002_sub.las
5 2018-10-17T12:23:20.575: Wrote subset LAS file:
6 2018-10-17T12:23:20.575: D:\Temp\CRCST_Lidar\Working\RPS\Calibration site data\LAS\grid_17_100\cld_170720_01_040211_004_sub.las
7 2018-10-17T12:23:28.641: Wrote subset LAS file:
8 2018-10-17T12:23:28.641: D:\Temp\CRCST_Lidar\Working\RPS\Calibration site data\LAS\grid_17_100\cld_170720_01_040551_005_sub.las
9 2018-10-17T12:23:52.799: Creating project:
10 2018-10-17T12:23:52.799: D:\Temp\CRCST_Lidar\Working\RPS\Calibration site data\LAS\grid_17_100
11 2018-10-17T12:23:50.896: Running task PointCloudFeatureExtraction:

```

10.8 Project Budget and Expenditure

Project Budget:

Item	Quarter (\$) commencing 1 March 2017				
	Q1 Mar17	Q2 Jun17	Q3 Sep17	Q4 Dec17	Q5 Mar18
<i>Salaries</i>					
<i>Research Fellow/Assistant (Software development) (1 FTE)</i>	12,500	37,500	37,500	37,500	25,000
<i>Travel</i>					
<i>Melbourne/Brisbane travel, likely 4-6 trips required by PLs and research team</i>	500	1,500	1,500	1,500	1,000
<i>Equipment</i>					
<i>Computing expenses & consumables</i>	333	1,000	1,000	1,000	667
<i>Quarterly Totals</i>	13,333	40,000	40,000	40,000	26,667
<i>Total Budget</i>	\$160,000				

Project Expenditure	
ESRI + Harris Geospatial Consulting Services	\$130,660.00
Expenses for data capture by RPS	\$8,000.00
Travel (Sudarshan to Melbourne)	\$1,117.31
Total Expenditure	\$139,777.31
Unclaimed	\$20,222.69

DNRME staff In-kind contribution (Hours)	
Russell Priebbenow – Project Lead	255
Sudarshan Karki – Lead Researcher	2307
Govinda Baral – Research Assistant	131
Garry Cislowski – GPS Field Survey	15
Total Hours	2708

10.9 Project Timeline

Tasks, Milestones & Deliverables	6.5 week blocks									
	1	2	3	4	5	6	7	8	9	10
Literature review of R&D in utilisation of imagery and Lidar for cadastral data upgrading										
Formulation of cadastral upgrade requirements & consideration of any legal/administrative/technical constraints										
Investigation of accuracy aspects of feature extraction for cadastral feature measurement via imagery & Lidar										
(a) Identification of test areas for pilot project and (b) sourcing/preparation of imagery and Lidar data										
The development of processing pipeline & software to support manual extraction of cadastre-relevant features from the imagery and Lidar data										
Conduct tests of manual feature extraction, accuracy validation & assessment of cadastral upgrading feasibility & prepare interim report										
Further development of processing pipeline & software to support automated feature extraction of cadastre-relevant features from the imagery and Lidar data, along with processes for automatic cadastre upgrading										
Experimental evaluation of the semi- and fully automatic feature extraction and cadastral upgrading methodology developed, over a number of different sites										
Analysis of results of pilot study, with a focus upon both accuracy and the practicability and reliability of the developed cadastral										

Tasks, Milestones & Deliverables	6.5 week blocks									
	1	2	3	4	5	6	7	8	9	10
upgrading methodology										
Preparation of final report and recommendations from the cadastral upgrading project, including assessment of feasibility for operational implementation of project developments										■