UCGE Reports
Number 20187

Department of Geomatics Engineering

Automated Extraction of Digital Terrain Models, Roads and Buildings Using Airborne Lidar Data

(URL: http://www.geomatics.ucalgary.ca/links/GradTheses.html)

by

Yong Hu

October 2003

THE UNIVERSITY OF CALGARY

Automated Extraction of Digital Terrain Models, Roads and Buildings Using Airborne Lidar Data

by

YONG HU

A DISSERTATION

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

OCTOBER, 2003

UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled “Automated Extraction of Digital Terrain Models, Roads and Buildings Using Airborne Lidar Data” submitted by Yong Hu in partial fulfillment of the requirements for the degree of doctor of philosophy.

_____

Supervisor, Dr. Vincent Tao, Geomatics Engineering

_____

Co-supervisor, Dr. Michael Collins, Geomatics Engineering

_____

Dr. Naser El-Sheimy, Geomatics Engineering

_____

Dr. Isabelle Couloigner, Geomatics Engineering

_____

Dr. Nigel Waters, Geography

_____

Dr. Ahmed Mohamed, Alberta Research Council Inc.

October 3, 2003
Date

ABSTRACT

Airborne lidar has become a commercially viable remote sensing platform, and can provide accurate elevation data about both topographic surfaces and non-terrain objects. Its capability of mapping topography and 3-D models of civil objects is uncommon to other remote sensing technologies. This dissertation presents a collection of algorithms developed for automatically extracting useful information from lidar data exclusively. The algorithms focus on automated extraction of DTMs, 3-D roads and buildings utilizing single- or multi-return lidar range and intensity data. The hierarchical terrain recovery algorithm can intelligently discriminate between terrain and non-terrain lidar points by adaptive and robust filtering. It processes the range data bottom up and top down to estimate high quality DTMs using the hierarchical strategy. Road ribbons are detected by classifying lidar intensity and height data. The 3-D grid road networks are reconstructed using a sequential Hough transformation, and are verified using road ribbons and lidar-derived DTMs. The attributes of road segments including width, length and slope are computed. Building models are created with a high level of accuracy. The building boundaries are detected by segmenting lidar height data. A sequential linking technique is proposed to reconstruct building boundaries to regular polygons, which are then rectified to be of cartographical quality. Then prismatic models are created for flat roof buildings, and polyhedral models are created for non-flat roof buildings by the incremental selective refining and vertical wall rectification procedures. Many attributes of these building models are derived from the lidar data. These algorithms have been tested using many lidar datasets of varying terrain type, coverage type and point density. The results show that in most areas the lidar-derived DTMs retain most terrain details and remove non-terrain objects reliably; the road ribbons and grid road networks are sketched well in built-up areas; and the extracted building footprints have high positioning accuracy equivalent to ground-truth data surveyed in field. A toolkit, called Lidar Expert, has been developed to bundle these algorithms and to offer the capability of performing fast information extraction from lidar data.

ACKNOWLEDGEMENTS

First and foremost, I must thank my parents for their great support and understanding to pursue a goal lasting over thirty years. The fourteen years of studying at same schools and the same university with my little brother are always priceless. I would like to thank God for life itself. All that I have is due to his grace, and I would give glory to him.

I would thank all those who have helped me in my work here at University of Calgary and York University: Drs. Vincent Tao, Yang Gao, Rod Blais, Michael Collins, and Ms. Lu-Anne Markland. I have learnt a great deal in my time here at University of Calgary, but nothing more important than the values and principles they taught me. Special thanks go to my former advisor, Mr. Zhifeng Tang, for his great help.

I also would thank my fellow graduate students and colleagues in this department and in earth science department, York University, who have made my lives colorful. They are Mr. Zhizhao Liu, Kongzhe Chen, Drs. Longmei Li, Quanke Wang, Chaowei Yang, and Mr. Zhe Liu, Jihong Zhang, Jiun Han Keong, Xiangqian Liao, Junjie Liu, Chaochao Wang, Wentao Zhang, Changlin Ma, Ming Zhong, Chris Wang, Jeff Xu, and Ivan Wang.

I would like to thank my classmates in the past fourteen years: Mr. Chenyin Zhong, Ms. Yanfen Le, Drs. Yunfeng Jian, Hao Zhao, Chuanzhong Xu, Xiangru Yuan, Wenlong Zheng, and Wanshou Jiang, who have helped me a great deal till today. There are many other people who have helped and encouraged me in my work and life whom I have not mentioned. I want to thank them all for all they have done for me.

Support from Talbert Abrams Grand Award (ASPRS, 2003), The University of Calgary Silver Anniversary Graduate Fellowships (U of C, 2001~2003), Robert E. Altenhofen Memorial Scholarship Award (ASPRS, 2002), Graduate Research Scholarships (U of C, 2000~2001), and Research Assistantships from University of Calgary (1999~2001) and York University (2002~2003) are gratefully appreciated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AEM	Airport Elevation Model
AID	Airfield Initiative Document
ALS	Airborne Laser Scanning, or Airborne Lidar System
ASCII	American Standard Code for Information Interchange
CE	Circular Error
CIR	Color Infrared
COTS	Commercial Off-The-Shelf
CSG	Construction Solid Geometry
DEM	Digital Elevation Model
DGPS	Differential Global Positioning System
DIAL	Differential Absorption Lidar
DLG	Digital Line Graph
DN	Digital Number
DNM, DNTM	Digital Non-terrain Model
DOQ	Digital Orthorectified Quadrangle
DSM	Digital Surface Model
DT	Delaunay Triangulation
DTM	Digital Terrain Model
EHT	Enhanced Hough Transformation
FN	False Negative
FP	False Positive
GCP	Ground Control Point
GIS	Geographical Information System
GRM	Grid Road Model
GRN	Grid Road Network
GPS	Global Positioning System
GSD	Ground Sampling Distance

HT	Hough Transformation
HTRA	Hierarchical Terrain Recovery Algorithm
IDW	Inverse Distance Weighting
IfSAR, InSAR	Interferometric Synthesized Aperture Radar
IMU	Inertial Measuring Unit
INS	Inertial Navigation System
ISODATA	Iterative Self-Organization Data Analysis
LE	Linear Error
LIDAR, LADAR	LIght/LAser Detection And Ranging
LRF	Laser Range Finder
MBR	Minimum Bounding Rectangle
MDL	Minimum Description Length
MSL	Mean Sea Level
NAD27	North American Datum of 1927
NDVI	Normalized Difference Vegetation Index
OIS	Obstruction Identification Surface
POS	Positioning and Orientation System
RCM	Road Centerline Model
RI	Risk Index (for pipeline or airfield obstructions)
ROI	Region of Interest
RMSE	Root Mean Square Error
SAVI	Soil-Adjusted Vegetation Index
SCC	Stress Corrosion Cracking
SHT	Sequential Hough Transformation
SLT	Sequential Linking Technique
TIN	Triangulated Irregular Network
TP	True Positive
VHM	Vegetation Height Model
VSM	Vegetation Support Model
WGS84	World Geodetic System of 1984
w.r.t.	With Respect To

LIST OF SYMBOLS

(X, Y, Z)	Three coordinates of lidar points in 3-D Cartesian space $\Re^3$
(w, y, k)	Three rotation angles: roll, pitch and yaw
(x, y)	Two coordinates of a point in 2-D Cartesian space $\Re^2$
(i, j)	Integer row and column coordinates of a pixel in a digital image
M, N	Numbers of rows and columns of a digital image
$I(x, y), I(i, j)$	Continuous and discrete image functions
$N_8(p), N_4(p)$	8- and 4-neighborhoods of a pixel p
S_t	Translation operation of S by a vector $t \in \Re^2$,
$A \oplus S$	Morphological dilation of set A and the structuring element S
$A \Theta S$	Morphological erosion
$\breve{S}$	Reflection operation
$A \circ S$	Morphological opening
$A \bullet S$	Morphological closing
exp(.)	Exponential function
$G(x, y)$	2-D Gaussian function
$I * G$	Convolution of an image I and an operator G
$E(x, y)$	Edge strength in an image
$b(x, y)$	Gradient or normal direction in an image
E_{snake}	Energy function of active contour models
arctan(.)	Inverse tangent function
log(.)	Logarithm of base 2
C_l	A priori covariance matrix of the observation vector
n_{ROI}	Number of points in lidar dataset
A_{ROI}	Area of the study region
s_c	Size of grid cell of an image
s_w	Size of the largest filtering window

s_{gap}	Size of void areas to be filled
tan(.)	Trigonometric tangent function
Th, Th_1, Th_2	Thresholds related with the smooth condition
ln(.)	Natural logarithm
ΔR	Range resolution of laser
$\wedge$, $\vee$	Logical AND and logical OR
Z_x', Z_y'	First-order partial derivatives of surface
Z_x'', Z_y''	Second-order partial derivatives of surface
e	Positive zero threshold
s_{DTM}^2	Noise variance of the DTM
s_b, s_b'	Small and large direction tolerance threshold
$p(x, y, b)$	Oriented edge point
$d(p_u, p_v)$	Euclidean distance between two points
r, q	Distance and angle parameters of a Hough line
$l(r,q)$	Hough line
$d(p, l)$	Distance between a point and a Hough line
s_r, s_q	Distance and angle tolerance thresholds
B_o	Building boundary consisting of a closed chain of orientated pixels
n_G, n_B	Number of points in the generalized and original boundaries
s	Edge segment associated with oriented edge points
$d(s_i, s_j)$	Euclidean distance between two edge segments
$\bar{x}$, s^2	Sample mean and sample variance
$N(a, s^2)$	Normal distribution with the mean a and the variance s^2
$c^2(m)$	Chi-square distribution with m degrees of freedom
$t(m)$	Student distribution with m degrees of freedom
T_{c^2}, T_t	Test statistics
$l_{\max}$	Maximal eigenvalue of a matrix

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

Topographic data are critical for most uses of information representing the true terrain surface. Mapping of flood plains and flood-prone areas, pipeline corridor planning, canal or highway engineering, and basic topographic mapping all require elevation data for the terrain surface itself, at varying levels of accuracy. Maps represent important media to interpret and communicate the environment surrounding us. The geomatics community has constantly taken advantage of emerging technologies that offer higher positioning accuracy, or are faster and cheaper. The process of collecting topographic data over the past thirty years has seen the mapping industry move from brute force approaches (e.g., field surveying) to passive sensing approaches (e.g., photogrammetry and remote sensing) and, more recently, to active sensing approaches (e.g., lidar and IfSAR).

Airborne lidar is relatively new technology complementary to traditional field surveying and photogrammetry approaches. In order to expedite map production from lidar data, great effort has been made to minimize the need for expensive manual digitizing and to automate the extraction of these features carrying attributes as expected in GIS. However, lidar collects data from the first surface with which laser beams interact. The resulting DSMs are representative of the elevation of that surface composed of both the “bare Earth” surface and aboveground features. While this first surface can be useful for applications such as landscape analysis, environmental study or flight simulation, it does not provide elevation data required by most users to detect the bare Earth surface.

The bare Earth or ground surface is defined as the continuous and smooth surface that has nothing visible below it (Haugerud and Harding, 2001), and is represented by the DTM in civil and surveying engineering communities. The bare Earth surface is composed of the topsoil or thin pavements (e.g., roads), while the aboveground features include vegetation

and artificial objects that are classified as detached objects and attached objects. The detached objects (e.g., buildings and trees) rise vertically on all sides above the bare Earth; while attached objects (e.g., bridges and viaducts) rise vertically above the bare Earth only on some sides but not all (Sithole and Vosselman, 2003).

Photogrammetry has been a standard approach for the production of DTMs or DEMs for several decades. Photogrammetric models are formed in stereo-plotters using interior, relative and absolute orientation procedures (Mikhail et al., 2001; Egels and Kasser, 2001). Stereo-compilers manually digitise geomorphic features called hard breaklines such as drainage channels, road edges, and bottoms of ditches in one layer. Undulations in topography are digitised by so called soft breaklines. Then spot heights are added up at regular intervals manually by keeping the floating mark on the ground. Later, the DTM is generated from these breaklines and spot heights using interpolation techniques (ISM, 1997). The digital or softcopy photogrammetric systems first create DSMs by matching conjugate points using epipolar resampled stereo images (Greve, 1997; Mikhail et al., 2001; Egels and Kasser, 2001; Tao, 2002). The DSM is then adjusted by using a combination of manual and automatic techniques that look between surface objects such as trees or buildings all the way to the ground, where possible, to generate the DTM. The contour map becomes a cartographic by-product that provides an impression of the terrain (ISM, 1997). To date, these contours have been the standard source of information for topographic analyses.

In the past, the mapping industry viewed lidar technology with great suspicion as either too complicated, or too expensive. While lidar can provide a very detailed first surface model, the large number of points becomes a concern in production. What is worse, the lidar data could not readily be processed using existing COTS photogrammetric systems. Today, this technology is mainly accepted as a critical monitoring tool by powerful geospatial information users that include mapping and disaster management agencies, oil and gas exploration companies, telecommunications industry, pipeline companies, and environmental agencies (ALM, 2003). Compared with the mature state of lidar systems, the processing of lidar data is still in an early phase of development (Ackermanm, 1999;

Flood, 2001; Fowler, 2001). Despite the capabilities of direct georeferencing, the largest barrier of accepting lidar as a mapping tool has been the limited capability of automated extraction of cartographic features (e.g., DTMs, roads and buildings) from lidar data.

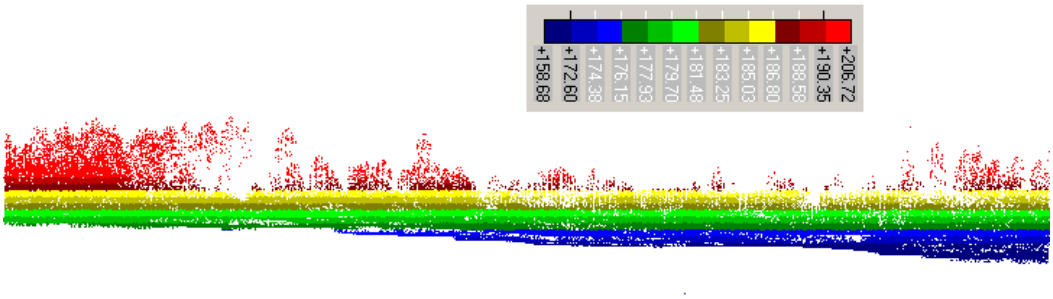
Lidar, IfSAR and photogrammetry have their own advantages and disadvantages, and their integration may be needed to improve the terrain modeling and to generate cartographically accurate contours for large-scale topographic maps (Thompson and Maune, 2001; Mercer, 2001; Fowler, 2001; Molander et al., 2002; Veneziano et al., 2002; Satale and Kulkarni, 2003). For example, breaklines along roads, bridges, overpasses and drainages have to be collected by stereo-compilers using high-resolution stereo images, and mass points may be added in areas devoid of lidar points. Mapping firms should leverage the strength of these technologies to create a product that meets accuracy, cost and schedule requirements in order to maximize benefits.

1.2 AIRBORNE LIDAR TECHNOLOGY

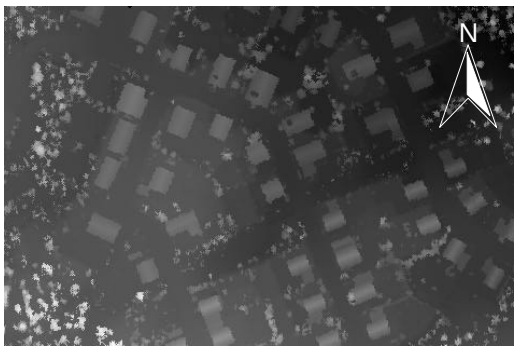
Airborne lidar or ladar represents a promising and relatively new technology for the highly automated acquisition of DSMs with high accuracy comparable with traditional land surveys and photogrammetry, but faster, denser and more economical (Fowler, 2001; Flood, 2001). The first experiments using lasers to measure distances to the Earth's surface began in the 1970s. This technology has evolved into two distinct lidar sensors: small footprint, time-of-flight laser altimetry, and large footprint, waveform-digitizing technique that analyses the full return waveform to capture a complete elevation profile within the target footprint (Flood, 2001). This thesis only discusses the processing of data acquired by small footprint range finders (see Appendix A).

A lidar system is often mounted in an aircraft, and uses a laser scanner, a geodetic-quality GPS receiver and an INS unit (see Figure A.1). It is an active, airborne sensor that works by rapidly firing laser beams downwards the ground and measuring the time the light takes to reflect from the ground objects and to return to the laser scanner (Wehr and Lohr, 1999). Modern lidar systems are capable of scanning large areas quickly and producing

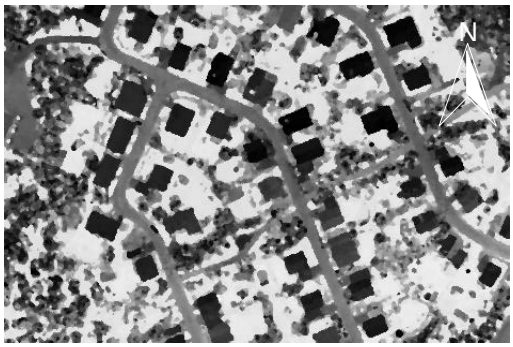
accurate point clouds with planimetric accuracy within 1 m and vertical accuracy around 15~20 cm (see Table A.1). Besides range data, many lidar systems collect the intensity of the reflected signal, multiple (up to five) returns for each pulse, and images taken by digital cameras (Mohamed et al., 2001; Optech, 2003).



(a) The cross-section of raw lidar points (south to north looking)



(b) Range image



(c) Intensity image

Figure 1.1. A sample lidar dataset (Village site)

A sample first-return lidar dataset collected at a village is shown in Figure 1.1, where the cross-section is coloured by elevation and the range and intensity images have a GSD of 0.6 m. The amount of information contained in such high-density 3-D point clouds is enormous. A number of natural and manmade features, such as bare topsoil, tree, road, building, waterway, power line, bridge and ramp, are all easily discernable to the human eye in cross sections and range and intensity images. However, computers cannot perform visual interpretations like the human eye to discriminate these features. Extracting the above features using lidar data not only is of great practical importance for engineering applications but also provides an excellent testbed for developing techniques for understanding complex scenes (Tao and Hu, 2001).

1.3 ISSUES IN AUTOMATED INFORMATION EXTRACTION

The early efforts of research and development of lidar technology have been placed on the improvement of lidar systems in terms of system performance, positioning accuracy and sensor calibration. Although mapping applications using lidar data are of great importance for largely developing this market, currently no COTS software packages are satisfactory for automated information extraction from lidar data (Flood, 2001; Tao and Hu, 2001). Most modeling tasks are done by human operators, and the production procedures depend heavily on manual operations, which are very time-consuming.

Lidar data can be used to generate DTMs, which are important for many applications such as terrain analysis and hydrology. Extracting thematic features, including road, building and vegetation, from these 3-D point clouds are paid much attention recently. However, lidar data provides coordinates only, and thus its interpretability is limited at many aspects (Ackermanm, 1999). The processing of lidar data is still in an early phase of development although lidar systems are already in a mature state (Ackermanm, 1999; Flood, 2001, ALM, 2003). Existing algorithms often exploit only part of information contained in lidar data, or just focus on processing specific scenes, for example, urban or forested areas (Tao and Hu, 2001). In addition, researchers often reported processing results using relatively small datasets.

The penetrable vegetation or solid surfaces can be detected if multi-return lidar range data are available (Kraus and Rieger, 1999). Laser beams are capable of penetrating foliage especially in the off-leaf condition (Sapeta, 2000; Optech, 2003). The differences between the first and last pulses at the same spots show how far laser pulses may penetrate into the vegetation on a point-to-point base rather than for a whole area, and thus provide a clean map indicating the existence of tall vegetation (Elberink and Maas, 2000; Zhan et al., 2002b; Hu and Tao, 2003; Alharthy and Bethel, 2003).

DTM generation is among one of the most direct applications for lidar data processing. Because the sampled lidar points are collected from the top of the Earth’s surface partly

covered by aboveground features, the DTM generation needs to identify the terrain points on the bare Earth, and to remove non-terrain points hit on trees, buildings and other constructions based on terrain points. In academic community, many algorithms have been developed to process relatively flat, urban, forested, mountainous areas, or in a few cases, hybrid areas (Weidner and Forstner, 1995; Kilian et al., 1996; Hug and Wehr, 1997; Kraus and Pfeifer, 1998; Petzold et al., 1999; Vosselman, 2000; Axelsson, 1999; Axelsson, 2000; Kraus and Rieger, 2000; Wang et al., 2001; Tao et al., 2001; Pfeifer et al., 2001; Elmqvist, 2001; Schickler and Thorpe, 2001; Lee and Schenk, 2001; Fraser and Jonas, 2001; Sithole, 2002; Brovelli et al., 2002; Masaharu and Ohtsubo, 2002; Rabor et al., 2002; Schiewe, 2003; Hu and Tao, 2003; Sithole and Vosselman, 2003). However, these algorithms usually use single-return range data only, and omit the valuable information contained in multi-return range data or intensity data.

The bare Earth may be estimated with low quality in complex areas due to various combinations of terrain type, coverage type and point density. For example, algorithms designed to process forested areas likely perform poor when processing urban areas (Kraus and Pfeifer, 1998), or vice visa (Wang et al., 2001). Thus, researchers have to customize different versions of these algorithms for different situations. For example, Schickler and Thorpe (2001) and Pfeifer et al. (2001) have modified Kraus and Pfeifer's (1998) filtering method in several ways to process urban areas. In industry community, the adaptive filtering algorithm based on TINs was developed by Axelsson (2000), and has been implemented in the COTS software TerraScan ^{TM} from TerraSolid (2001). To my knowledge, this is the only algorithm that is widely used for practical production of DTMs. But, the production procedure needs interactive inputs from operators, and heavily depends on manual operations to generate high-quality DTMs.

Lidar range data is able to improve the analysis of optical images for detecting roads in urban areas (Hofmann, 2001). Lidar intensity data has good separability if the wavelength of the laser is suitable for ground materials (Song et al., 2002). Roads have homogeneous reflectivity in lidar intensity and the same height as bare surface in elevation. Initial results on classifying intensity and range data for the detection of road ribbons have been

given, but with low quality. That is, the roads are represented as only rough centerlines (Hu and Tao, 2003) or binary images (Alharthy and Bethel, 2003) instead of vectors. Most of the road extraction algorithms are quite different in nature due to the differences in data sources, assumptions about road models and human interaction (Laptev, 1997; Park and Saleh, 2001). Most of them focus on the extraction of free-form curvilinear roads. To process the special case of grid road networks in urban areas, Price (1999) applied a local grid constraint to reconstruct these street grids starting with an initial seed intersection that provides the size and orientation of the regular grid.

Lidar data contains much 3-D information about buildings. But buildings are highly unstructured segments with very complex contents and elevation variations in range data. Buildings can be located at any places, and may be surrounded by other objects with similar radiometric properties in intensity such as roads. The geometric resolution of lidar data may be limited due to irregular sampling patterns, and tall trees may occlude parts of building or house roofs in densely vegetated and built-up areas.

Many algorithms focus on the detection of building footprints because the reconstruction of complex building roofs in 3-D is very difficult (Baltsavias et al., 1995; Hug, 1997; Yoon et al., 1999; Zhan et al., 2002a). The use of lidar intensity or range data can benefit the separation of buildings and vegetation (Henricson et al., 1996; Brunn and Weidner, 1998; Elberink and Maas, 2000; Hofmann, 2001; Hu et al., 2003). But building footprints cannot be detected fully automatically and reliably, and are often assumed to be of simple shapes with orthogonal corners such as rectangles or low-quality polygons (Weidner, 1995; Vosselman, 1999; Wang and Schenk, 2000; Vestri and Devernay, 2001). The reconstruction of 3-D building models is more difficult and is often limited to simple and specific cases assuming rectilinear footprints, parametric shapes, flat or symmetric sloped roofs (Weidner and Forstner, 1995; Lin and Nevatia, 1998; Maas and Vosselman, 1999; Vosselman, 1999; Wang and Schenk, 2000; Vosselman and Dijkman, 2001; Elaksher and Bethel, 2002). Since the building detection is not reliable enough, extra data such as ground plans have to be used to exactly define the building boundaries (Haala and Anders, 1997; Lemmens et al., 1997; Haala et al., 1998; Haala and Brenner, 1999; Brenner, 2000;

Vosselman and Dijkman, 2001). The reported results are often a few separate or small groups of buildings (Weidner and Forstner, 1995; Maas and Vosselman, 1999; Wang and Schenk, 2000; Elaksher and Bethel, 2002). To date, 3-D building models for the use in GIS are mainly produced by semi-automatic procedures (Rottensteiner, 2001).

1.4 RESEARCH OBJECTIVES AND METHOGOLOGY

1.4.1 Research Objectives

The results of this work will serve the increased demand of automation for 3-D information extraction using remotely sensed large datasets. The research is aimed to develop and improve the capabilities of automated feature extraction from lidar data by addressing following questions:

- How much information can be exploited from lidar data by analysing its geometric and radiometric properties, which are unique compared to other remote sensing data sources?
- How can the combination and fusion of single- or multi-return lidar range and intensity data be used to efficiently exploit different feature information?
- How many cartographic features can be extracted automatically by applying computer vision and digital photogrammetry techniques?
- How the automation of feature extraction can be achieved or improved by applying effective strategies or constraints?

The scope of the research is centred on developing and implementing intelligent algorithms for automated extraction of three major types of cartographic features for scenes with varying terrain and coverage types. The automation, robustness and performance of these feature extraction algorithms are particularly emphasized. Employment of effective processing strategies to improve the automation is a key to the implementation of these algorithms. There are three problems tackled in this research:

- Generate DTMs for complex scenes using either single-return range data or multi-

return range and intensity data for large datasets.

- Detect road ribbons for complex scenes, and reconstruct grid road networks in urban areas, using both intensity and range data.
- Detect buildings in urban area, and reconstruct prismatic or polyhedral building models for large datasets, using multi-return range and intensity data.

It is expected that the developed methods be of practical value to and impact in improving the automation capabilities in feature extraction from lidar data.

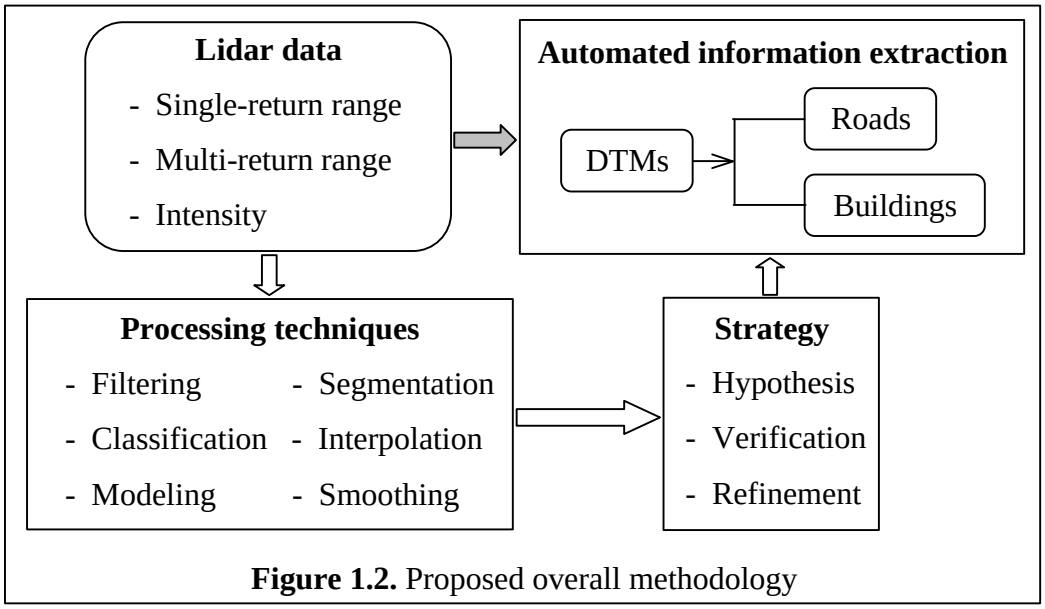


Figure 1.2. Proposed overall methodology

1.4.2 The Methodology

Figure 1.2 illustrates subsequent processes proposed in this thesis and the interrelations among the major components for automated information extraction from lidar data. The proposed components form a general framework, in which three types of major cartographic features are exploited. After analysing the geometric and radiometric characteristics of lidar range and intensity data, I intend to focus on the extraction of DTMs, roads and buildings, using single- or multi-return lidar range and intensity data.

Based on extensive and critical literature review, the problems in current feature extraction algorithms are analyzed. I will improve those algorithms by combining related data processing techniques or developing new techniques under the hypothesis and

verification paradigm. The extracted features may be further refined according to specific assumptions or constraints about them. In any event, DTMs are generated first to describe the bare Earth. The reliable extraction of roads and buildings needs to know the height information relative the bare Earth that has removed the terrain relief. This height data puts all the features on a flat reference plane, and can be obtained by subtracting the derived DTM from the lidar DSM. These algorithms will be tested for complex scenes.

1.5 CONTRIBUTIONS

The following contributions are included to achieve the above objectives:

- [1] I developed the hierarchical terrain recovery algorithm, which builds a novel technical framework of data-driven DTM generation for single- or multi-return lidar data. I proposed to integrate information about terrain relief, slope and data density in a new formula of the smooth condition to identify non-terrain points based on hypothesized terrain points (i.e., local minima) in a hierarchical approach. Thus, a single algorithm is suited to efficiently process different scenes of varying complexities. I proposed to use four evidences to identify road points, which add denser terrain points. I also proposed to adaptively adjust the parameters in the smooth condition by assimilating information about vegetation and roads.

- [2] I proposed to detect road ribbons by classifying multi-return intensity and height information and generate road centerline models by applying morphological operations and image thinning. In urban areas where roads are in a regular grid layout, I proposed a new model-driven method to automatically reconstruct 3-D grid streets by applying a global grid constraint and making hypothesis verification based on road segments. The grid road model in vector format is expressed as road crossings and segments holding the connectivity topology. The width of a road segment is also calculated automatically.

[1] This algorithm has been discussed briefly in Hu and Tao (2003).
[2] These algorithms have been discussed briefly in Hu and Tao (2003). The detection of road ribbons using multi-return intensity and height data is developed independently in Alharthy and Bethel (2003).

- [3] I proposed a data-driven workflow to automatically reconstruct complex building boundaries as rectangles, quadrangles, or regular polygons. I first detect building boundaries by classifying object boundaries using shape measures. Then I applied the new algorithms, including the constrained searching, enhanced Hough transformation and sequential linking technique, all follow the hypothesis verification paradigm and an optional refinement. I proposed to classify building roofs into flat and non-flat classes by statistical testing. Finally I used the selective refinement technique to model non-flat roofs and rectify building walls by applying the vertical wall constraint, which is used to define three new types of ridge vertices.

1.6 OUTLINE OF THE THESIS

This thesis is divided into six chapters and three appendices that are organized following the proposed framework shown in Figure 1.2. An overview of these chapters and appendices is given in this section.

Chapter 1 briefly introduces airborne lidar technology, reviews information extraction status using lidar data, and discusses research objectives.

Chapter 2 gives a literature review on related processing techniques and representative algorithms on the extraction of DTMs, roads and buildings focusing on using lidar data.

Chapter 3 discusses a hierarchical terrain recovery algorithm for data-driven DTM generation. The methodology for processing single-return range data is outlined first, and is then modified by assimilating road or vegetation information derived from intensity data or multi-return range data. The algorithm has four major steps: generation of image pyramids, establishment of the smooth condition, identification of topographic points, and interpolation at non-terrain points. Finally DTMs are refined by correcting problematic areas and smoothing non-terrain undulations.

[3] These algorithms have been discussed in detail in Hu et al., (2003).

Chapter 4 presents the detection of road ribbons by classifying both intensity and height data. The emphasis is given to the model-driven reconstruction of 3-D grid streets in urban areas by applying a global grid constraint and utilizing the road segment based hypothesis verification. The attributes of road segments, including width, length and slope, are also calculated automatically.

Chapter 5 deals with a data-driven workflow for building detection and reconstruction from lidar data. Building boundaries are divided into groups of edge segments using the enhanced Hough transformation, and the edge segments are permuted in correct order using the sequential linking technique to form regular polygons. Finally the flat and non-flat roof buildings are modeled as prismatic and polyhedral shapes, respectively.

Chapter 6 summarizes the contributions of the research work, and recommends research topics of future work.

Appendix A gives a detailed description of airborne lidar technology and its applications.

Appendix B contains general knowledge about image processing, computational geometry, numerical analysis and quality measures.

Appendix C demonstrates three applications of the information extracted from lidar data using the proposed algorithms.

CHAPTER TWO

LITERATURE REVIEW

The task in photogrammetry and remote sensing community is to collect data about the Earth's surface and derive an interpretation to complete certain mapping or engineering projects. Data sources to support such tasks are increasingly available from a wide variety of platforms and sensors. The primary data sources include optical and radar images, IfSAR data and lidar data. Many techniques have been developed to interpret lidar point clouds into useful representation for many applications such as 3-D landscape modeling.

Airborne lidar systems use accurate direct geo-referencing technology and collect the surface elevation data digitally (Burtch, 2002; Optech, 2003). Lidar DSMs are composed of point clouds returned from the top of the Earth's surface that is partly covered by manmade and natural ground objects such as trees and buildings (Ackermanm, 1999; Flood, 2002). In addition to range data, some lidar systems collect information on the intensity of the reflected signal and information of multiple returns for each pulse. The high point sampling densities, fast turn-around time and lower cost are among the most attractive characteristics for various mapping applications. In recent years, lidar data are rapidly becoming the standard and preferred data source, and has triggered numerous investigations into feature extraction that may be difficult or expensive when using traditional remote sensing data (Hill et al., 2000; Fraser and Jonas, 2001; Fowler, 2001; Tao and Hu, 2001; Veneziano et al., 2002; ALM, 2003).

"Lidar range data provides coordinates and coordinates only. On one hand, this allows fast and highly automated data processing. On the other hand, the interpretability of data is limited because no object information is provided because laser scanning is not capable of any direct pointing to particular objects or object features" (Ackermanm, 1999). For example, buildings, dense tree stands, dam walls, bridges and large rock outcroppings may have similar elevation cross-sections. Ancillary data have to be used to develop a general post-processing approach, such data are intensity image, multi-spectral or hyper-

spectral imagery, and GIS data such as ground plans (Tao and Hu, 2001). The systematic integration of lidar data and optical imagery data will constitute an effective fusion with photogrammetry, from a methodical and technological point of view (Ackermanm, 1999). In this way, the integration with photogrammetry would be extended to geospatial applications in a wide range (Thompson and Maune, 2001; Molander et al., 2002).

Over the past ten years, many techniques have been developed to interpret and model lidar collected irregular 3-D point clouds into useful representations for use in GIS as partly reviewed in Tao and Hu (2001). Some techniques obtain more cues by integrating lidar data with other data sources to ease certain processing. For example, lidar intensity data can be used for visualization of the surface, and to improve classification of objects in combination with the range data (Hoffman, 2001; Hu and Tao, 2003; Alharthy and Bethel, 2003). The DTM generation is the most direct application of lidar data. Since the lidar beams are hit on the top of the Earth's surface, removing aboveground obstructions such as trees and buildings in the DSM is one of the most critical steps. Automated analysis and classification of various application-specific features (e.g., road and power lines) are also of particular interests and are paid more attentions recently.

The software tools available to the entire community will be one of the most significant areas of change in lidar industry (Flood, 2001). To date, the vast majority of lidar data processing is conducted using proprietary software developed independently by researchers, data providers, or provided by the sensor manufacturer to its clients but not available as separate packages, including Optech's REALM, SAIC's LIDAR Toolkit, Sanborn's FASE. "This situation presents a significant barrier to end users as lidar processing is presented as a black box with limited insight into the actual manipulation of the data and a very limited ability of the end user to recreate, reclassify, manipulate or modify the datasets. The fact that few of these proprietary algorithms are published is also a concern among researchers. Currently, there are only a few software products (e.g., TerraSolid's TerraScan) on the market that can efficiently handle the large point densities generated by lidar sensors, but this situation is changing rapidly" (Flood, 2001). Third-party products specifically designed for visualizing and manipulating lidar point clouds

are starting to appear, such as APL's QT Viewer and Mensi's RealWorks Viewer.

This chapter organizes recent publications pertaining to information extraction from lidar data and ancillary remote sensing data by presenting an overview of these techniques developed toward highly detailed and accurate digital modeling for earth surface in 3-D. The algorithms are classified into several groups according to the basic subtasks, which are DTM generation, road extraction and building extraction. Representive examples of various methods are summarized for each subtask. The problems around the issues pertaining to these methods are also discussed.

2.1 RELATED TECHNIQUES

Existing airborne lidar technology does not provide a real-time solution to services; it needs a large amount of work in the post-processing stage (Ackermanm, 1999). There are many algorithms developed for information extraction from lidar data or optional ancillary data. Some algorithms are purely based on range data itself; others are based on range and ancillary data (Tao and Hu, 2001). Automatic or semi-automatic analysis and classification of various specific ground features (e.g., power line and railroad) are also being developed (TerraSolid, 2001). Most algorithms use a few basic techniques described below (Tao and Hu, 2001), some of which have no definitive boundaries until they are used in specific data processing procedures.

- *Filtering* usually means removing unwanted measurements, or finding ground surfaces from a mixture of ground and aboveground measurements (Petzold et al., 1999). To distinguish points located on buildings and tree canopies from those that are expected to be on the ground, order statistics and morphological filters or weighting functions are often applied (Weidner and Forstner, 1995; Hug and Wehr, 1997; Kraus and Pfeifer, 1998; Vosselman, 2000; Wang et al., 2001; Tao et al., 2001; Fraser and Jonas, 2001; Wack and Wimmer, 2002). Raw lidar point clouds may be processed to remove blunders, yet preserving the discontinuities between the surfaces (Umasuthan and Wallace, 1996; Zinger et al., 2002).

- *Segmentation* means the separation of a point cloud into homographic patches describing different geometric, radiometric or texture structures (e.g., road, building and vegetation) by applying thresholding, clustering, boundary detection or perceptual organization algorithms (Weidner, 1995; Ruskone, 1996; Lee and Schenk, 2001; Brovelli et al., 2002; Sithole, 2002; Schiewe, 2003; Hu et al., 2003; Hu and Tao; 2003; Alharthy and Bethel; 2003). Building segments can be detected by comparing elevation slices (Zhan et al., 2002a).
- *Classification* is used to discriminate among several categories of ground objects (e.g., buildings, roads, grasslands and trees) to group into different classes of point clusters by applying pattern recognition algorithms, such as ISODATA, Bayes classifier, and *k*-means (Henricson et al., 1996; Hug, 1997; Haala and Brenner, 1999; Elberink and Maas, 2000; Song et al., 2002; Alharthy and Bethel, 2003). Shape measures can be used to distinguish between vegetation and building boundaries (Parker, 1993; Wang and Schenk, 2000).
- *Interpolation* estimates the elevation at some location (Pfeifer et al., 2001; Wang et al., 2001). There are many methods to interpolate using the scattered points or a raster format surface. The most commonly used interpolation methods are IDW, TIN and Kriging (Axelsson, 2000; Zinger et al., 2002; Song et al., 2002). The most popular interpolation methods for raster images are nearest neighbor, bilinear interpolation and cubic convolution (El-Sheimy, 1998). Zinger et al. (2002) proposed an energy minimization approach to avoid drawbacks relevant to three well-known interpolation methods in resampling lidar data in urban areas.
- *Modeling* reconstructs the ground surface or an object based on geometric properties of its shape. The ground surfaces are approximated as piece-wise smooth patches or TINs (Axelsson, 2000; Elmqvist, 2001; Wang et al., 2001; Masaharu and Ohtsubo, 2002). Different combinations of basic road characteristics can be used to model roads in different resolution levels and contexts for different scene types by applying varying strategies (Garnesson et al., 1990; Vosselman and de Knecht, 1995; Steger et al., 1995; Barzohar and Cooper, 1996; Baumgartner et al., 1997; Laptev, 1997; Mayer et al., 1997; Fischler and Heller, 1998; Price, 1999; Couloigner and Ranchin, 2000; Hinz and Baumgartner,

2000; Park and Saleh, 2001). Objects are described in terms of their constituent shape features using parametric, prismatic and polyhedral models, which can be converted into wire frame or CSG models combining various primitives (Maas and Vosselman, 1999; Weidner and Forster; 1995; Wang and Schenk; 2000; Vosselman and Dijkman, 2001; Rottensteiner, 2001; Hu et al., 2003).

- *Smoothing* can remove random noise and produce a smoother surface so that the contours may look nicer. It is often an iterative process, comparing a point with nearby points and adjusting its elevation (Tao et al., 2001). Usually, a best-fit facet model is computed for a group of points, and the elevation of the centre point is adjusted to better match the facet (Besl and Jain, 1988; Wang et al., 2001).
- *Human interaction* is a mechanism to utilize human knowledge through human-computer interfaces. Fully automatic systems may fail in extracting features due to the complexity of the inverse engineering tasks and the shortages of lidar data (Hu et al., 2003). Instead, human operators can easily classify different terrain types (Sithole, 2002; Sithole and Vosselman, 2003), select seed points (Gruen and Li, 1994; Park and Saleh, 2001; Hu and Tao, 2003a), or delineate building positions and shapes (Brenner, 1999; Rottensteiner, 2001).

2.2 ALGORITHMS FOR DTM GENERATION

As lidar acquires first surface elevation data including non-terrain features, the core task first involves the separation of lidar returns that hit on the terrain surface from non-terrain returns. Data-driven techniques are often used for this task (Tao and Hu, 2001), and a few model-driven techniques are developed (Elmqvist, 2001; Brovelli et al, 2002). A number of algorithms have been reported in the literature, but most of them are not yet proved competent for industry production because of the complexity of the task and the need of much manual editing. Various combinations of terrain type (e.g., flat to mountainous), coverage type (e.g., low to high density vegetation, natural and cultural features) and (median to high) point density may cause DTM generation algorithms to fail in certain areas (Tao and Hu, 2001; Sithole, 2002; Sithole and Vosselman, 2003).

Aside from the systematic and accidental errors in lidar systems, the error sources influencing the quality of the final DTMs usually have four parts (Hu and Tao, 2003):

- *Type I error* results from the misclassification of terrain points as non-terrain ones.
- *Type II error* results from the misclassification of non-terrain points as terrain ones.
- *Representation error* results from the use of vector or raster data structures.
- *Interpolation error* results from the use of different data interpolation techniques.

The first two errors are the main sources resulting in a DTM of poor quality (Hu and Tao, 2003). Type I error discards true terrain points, and will lose details around the neighboring regions because no sufficient terrain points are identified. While Type II error attracts non-terrain points to participate in the DTM recovering, and this will definitely deteriorate the quality of the derived DTM around the neighboring regions. Obviously, Type II error will lead to worse results than the first one. Two factors resulting in Type I and Type II errors are analyzed in Sithole (2002). One factor is due to the characteristics of the scene, that is, the complex nature of terrain and arrangement of objects (e.g., buildings and vegetation); another factor concerns the characteristics of the data (e.g., point density, outliers, data gaps, etc.). These characteristics are then used to search for unclassifiable regions that cannot be classified with certainty. For example, the closer and larger an object is in relation to the terrain, the more difficult to separate it from the terrain. The classification of bridges and overpasses as terrain or non-terrain is not clear-cut, usually depending on specific applications (Sithole, 2002).

The last two errors are less important than the first two errors. The use of interpolation techniques depends on the use of data structures as described in Section 2.1. Some DTM generation algorithms process the raw lidar point clouds directly, while others process grid-based images. However, choosing a vector, a raster or a hybrid data structure would not result in significant differences in quality (Hu and Tao, 2003). Although it is expected to be able to achieve higher accuracy and scalability by processing lidar points directly, raster-based approaches have the extra advantage of allowing for using many image processing algorithms readily (Wack and Wimmer, 2002).

The identification of terrain or non-terrain points mainly relies on the local geomorphology. In moving filtering windows of varying sizes, local lowest points are usually the candidate terrain points, and local highest points are likely building roof or tree points. Weidner and Forstner (1995) presented a morphological filter based on gray values. It consists of an opening operator for extracting the DTM. In Kilian et al. (1996), morphological filters are used to eliminate non-terrain points, and then a weighted smoothing is performed based on the distance of lidar points to the opened surface. Hug and Wehr (1997) applied a morphological filter to separate terrain and non-terrain points by iteratively computing the probability values of each point to be a terrain point. In each window, the lowest point is identified and also other points that are higher within a certain range. Probability values are assigned to these identified points according to their relative height compared with the lowest points. That is, the lower the point, the larger probability to be a terrain point. Moreover, the lowest point in a large window has a larger probability than that in a small window to be a terrain point. Fraser and Jonas (2001) developed a recursive algorithm based on changes in the slope defined by points categorized as the terrain class. Four types of terrain type (i.e., flat, hilly, mountainous and steep mountainous) and three classes of vegetation (i.e., none, light and dense) are used to define twelve different morphological filters. These filters are decided upon from local site knowledge, aerial photography, maps, or from the examination of on-board video data.

Based on the fact that laser footprints often are on the treetops in wooded areas, Kraus and Pfeifer (1998) formulated an asymmetric weight function by statistical analysis to filter the lidar data. The algorithm is based on the linear prediction with an individual accuracy for each point, and works iteratively. “The surface is firstly computed with equal weights for the elevation values of all points, and thus runs in an averaging way between terrain points and vegetation points. Then the weights are re-calculated based on the residuals relative to the surface, and are used for the next computation of the surface” (Kraus and Pfeifer, 1998). Since the weight function is not symmetrical, this allows a sharper decline for values above its origin, and a slower decline or no decline at all for ground points. Points with large negative residuals have maximal weights and they attract the surface. Points with large positive residuals are eliminated. Above processing is repeated until a stable surface is obtained.

Petzold et al. (1999) proposed a filtering algorithm to distinguish points situated on buildings and on the vegetation from those expected to be on the ground. First, a rough terrain model is calculated using the lowest points found in a moving window of a rather large size that is based on the largest building size in region of interest. Then all points with a height difference exceeding a given threshold are filtered out and a more precise DTM is calculated. This step is repeated several times, reducing the window size and leading to the final DTM. The results are good, especially for wooded areas. The results are influenced by the final window size and the final threshold below which points are expected to be terrain points. A small window size or a large height threshold leads to misclassification of points on large buildings as ground points; a fairly large window size or a small height threshold removes small terrain discontinuities. Therefore, the parameters used depend on the terrain morphology and thus are different for flat, hilly and mountainous regions. "Breaklines are needed to avoid elimination of points that are situated on steep embankments along roads and railways or on steep slopes along ridges, or the geomorphologic structures are compiled in advance and used to achieve better results concerning the classification of ground points" (Petzold et al., 1999). The concept of finding lowest points in this algorithm has been the basis of many other algorithms (Axelsson, 2000; Wang et al., 2001; Masaharu and Ohtsubo, 2002).

Axelsson (1999) described an adaptive filtering method to process one scan line at a time. The basic idea is to start with a line beneath all the scan line points. This line is then connected to ground points from below, and the connection is controlled by the MDL principle. The MDL delimits the possible shapes and hence fluctuations of the resultant surface in some way. In Axelsson (2000), the above algorithm is implemented in 2-D to process dense urban areas where discontinuities may appear. A sparse TIN is created from seed points and densified in an iterative process. The TIN adapts to lidar points from below and is constrained in its curvature by data derived parameters. Since the seed points in the TIN constitute a subset of the actual lidar points, those true ground points will represent the real terrain surface with the accuracy equal to that of the lidar data. The main strength of the filter lies in the ability of handling surfaces with discontinuities, which is particularly useful in urban areas. This 2-D version has been implemented in TerraScan ^{TM} from TerraSolid (2001).

Masaharu and Ohtsubo (2002) developed a two-stage filtering method suitable for highly developed urban areas. The primary selection use the lowest point in regularly divided patches of the area, and a secondary selection is recursively applied to these points to remove points on building roofs and at the bottom of underground tunnels that may be remained in the primary selection. The secondary selection is based on whether the points in the primary selection are within one sigma from the mean of the neighboring lowest points of patches. If the point is judged out of the range by this statistical test, it is removed.

The sole use of lowest points often results in an under-estimated surface on average. Some algorithms explicitly use slope information to conquer the terrain relief in filtering. Vosselman (2000) developed a slope-based filter, which is proved to be equivalent to the erosion operator in mathematical morphology. In this approach, the ground is defined as points within a given slope range. The thresholds are determined by a stochastic approach that needs training. This slope-based filter was modified by Sithole (2001) to reduce Type I error happening on steeply sloped terrain. The modified method determines the thresholds with respect to the slope of the terrain. Schickler and Thorpe (2001) modified Kraus and Pfeifer's (1998) method in several ways. They use a triangulation of the surface instead of grid format, include independently measured mass points and breaklines in the estimation with appropriate weighting, add additional curvature constraints and slope constraints to control the shape, and employ the concept of surface classes to guide the estimation process.

Many researchers use a hierarchical strategy to assemble topographic details in a coarse-to-fine manner. Moreover, such processing algorithms are often robust and efficient, and are suitable for processing areas with complex terrain and coverage types. A hierarchical robust interpolation method has been implemented in SCOP++ (Pfeifer et al., 2001). This method actually extends Kraus and Pfeifer's (1998) filtering algorithm and applies it in a hierarchical manner to process built-up areas. It creates a data pyramid, filters the lower resolution data to generate a rough DTM, and accepts points within a certain tolerance band by comparing the derived DTM to the data of higher resolution. The latter two steps need iteration for each finer pyramid level. In Wack and Wimmer (2002), an approach based on the rasterization of lidar

points is presented. The algorithm consists of a hierarchical approach in combination with a weighting function for detecting non-terrain raster elements. The weighting function considers the terrain relief as well as the distribution within each raster element.

A surface fitting approach is developed to generate DTMs from DSMs acquired by the STAR-3i IfSAR system (Wang et al., 2001; Tao et al., 2001), and has been implemented in TerrainFit® by Intermap (Coleman, 2001). This method is designed to address open urban and rural areas of flat to mountainous terrain types. It does not address heavily forested areas. The approach consists of the creation of image pyramids, approximation of the DTM to the actual terrain, localization of problematic areas especially in mountainous regions, and the terrain smoothing. During the pyramid generation, non-terrain points are eliminated gradually, and the remaining points at the top level should be all true terrain points. Then a higher-level image is treated as the reference surface. At a lower-level image with higher resolution, multiple types of terrain points, such as points on the slopes and flat areas, are identified to retain the terrain details, and are used to interpolate a more accurate surface. This processing is repeated until the bottom level is reached. A self-diagnosis process is proposed to detect mountainous areas where the DTM is under-estimated because hills may be cut.

A number of algorithms based on segmentation (Hoover et al., 1996) and energy minimizing (Elmqvist, 2001) have been developed to segment gray images and close-range range data in computer vision for decades, and some of them have been extended to divide lidar data into homogeneous groups of different coverage types, including ground, building, tree, etc. Besl and Jain (1988) used the surface-curvature-sign to label eight fundamental surface types that correspond to eight classes of topographic points. That is each point on a piece-wise smooth surface can be characterized by the spatial properties of other points on the surface in l x l (e.g., l=7) neighborhoods surrounding the given point. However, the surface-curvature-sign labeling algorithm needs to fit a set of approximating functions (e.g., planar, bi-quadratic, bi-cubic and bi-quartic polynomials) incrementally to find the best form (a few seconds per fit on a VAX 11-780); then it calculates the mean and Gaussian curvature signs, which are viewpoint invariant, using the first- and second-order partial derivative estimates in image form.

Lee and Schenk (2001) used a 3-D perceptual organization of the point cloud into distinct surfaces. The method is intended to mimic the grouping of points in the way human perception would behave. Elmqvist (2001) developed an active shape model, which is an extension of the active contour model to higher dimensions, to estimate the ground surface by energy minimizing. The active shape model acts like a rubber cloth with elasticity and rigidity. The model is glued against the measured points from underneath with constraint forces, forming the envelop of the point cloud. The stiffness of the shape model stretches it out to a continuous surface between the ground points.

Brovelli et al. (2002) implemented a filtering algorithm based on spline interpolation and region growing techniques. This algorithm was designed for processing lidar data in urban areas, and has been implemented in the open source GIS - GRASS (see http://grass.itc.it/). The filter consists of five steps: preprocessing, edge detection, object region growing, problematic classification results correction, and interpolation. “It classifies multiple returns lidar data into terrain and non-terrain points. If points with uncertain classification remain, they are not used in the final DTM interpolation” (Brovelli et al., 2002). The validation of the filter was performed on an Italian dataset by means of topographic, geodetic and photogrammetric independent information, and presented good processing quality.

An alternative for conquering the difficulties in distinguishing terrain from non-terrain points solely using the first surface elevation data is to fuse auxiliary data such as multi-return lidar range and intensity data, or hyper-spectral images for land-cover classification. Kraus and Rieger (2000) presented a technique to separate wooded from non-wooded areas using multi-return range data to solve a special case. That is, the size of rock needles within the forested areas often does not exceed the size of a single tree, thus rock points are normally eliminated in the DTM along with vegetation points when using single-return range data only. The difference between first and last returns is of great help in that case since rocks exhibit differences close to zero in contrast to vegetation. The usage of aerial or satellite imagery may be of great help to better distinct between vegetation and non-vegetation.

In Hofmann (2001), the lidar range data is used to improve the analysis of high-resolution

images for detecting roads and buildings in urban areas. Rabor et al. (2002) developed an adaptive vegetation removal procedure to process lidar data mixed with multiple returns. They utilize a vegetation point removal algorithm in which the parameters are adaptively adjusted based on a vegetation map. The vegetation map is derived through the exclusive use of the lidar dataset, making the processing independent of ancillary data. Schiewe (2003) proposed a region-based and multi-scale approach. It consists of segmentation and follow-up fuzzy logic classification based on several features derived from DSM and multi-spectral images.

Sithole and Vosselman (2003) conducted a study to evaluate the performance of eight filtering algorithms. Seven characteristics, including data structure, test neighborhood, discontinuity, filter concept, single vs. iterative processing, replacement vs. culling and use of first pulse and reflectance data, are used to understand the behaviour of those filter algorithms. The filtering results are compared against reference data that are generated by manually filtering raw lidar data. It is found that in general the filters perform well in landscapes of low complexity. However, complex landscapes as can be found in city areas and discontinuities in bare Earth surfaces still pose challenges. It is suggested that future research be directed at heuristic classification of point clouds based on external data, quality reporting, improving the efficiency of filter strategies.

The quality evaluation of the generated DTMs is often based on comparisons against ground observations (Axelsson, 2000; Pfeifer et al., 2001; Fraser and Jonas, 2001), topographic maps (Kraus and Pfeifer, 1998), or reference DTMs (Axelsson, 2000; Wang et al., 2001; Brovelli et al., 2002), but has not been investigated systematically in the past. It is difficult to get the terrain elevations under buildings or directly measure them using surveying methods. The reference DTMs are usually interpolated from manually edited lidar points (Axelsson, 2000; Sithole and Vosselman, 2003), using propriety or expensive software tools. Pfeifer et al. (2001) reported absolute accuracies of derived DTMs for two datasets. The GCPs are measured using a GPS receiver or manually from photography. The vertical accuracy is from 8 cm to 11 cm RMSE for flat, sloped areas and railway ramp in Vaihingen dataset (relatively flat terrain), and is 10.5 cm RMSE for the overall area in Vienna dataset (flat terrain).

2.3 ALGORITHMS FOR ROAD EXTRACTION

Extraction of curvilinear features, especially roads, has been a popular research topic in the computer vision and remote sensing communities. During the last three decades, many approaches for road extraction have been developed in the literature. Most of them are quite different in nature due to the differences in available data sources, assumptions about road models, and human interaction (Laptev, 1997; Park and Saleh, 2001). These methods usually assume relatively simplistic road models and make little use of a priori knowledge. They are thus sensitive to disturbances like shadows, occlusions and varying resolutions. Park and Saleh (2001) made a comprehensive survey of automated and semi-automated extraction techniques of linear features from panchromatic and multispectral aerial and satellite imagery. The techniques are evaluated with respect to methodology, strengths, drawbacks, and implementation approach. The viability of hyper-spectral data is extrapolated for same purpose of utilization. The most important factors influencing the nature of a road extraction approach are the data sources available and the need of an operator to provide control information (Laptev, 1997).

The existing road extraction approaches utilize a wide variety of data sources largely to eliminate the uncertainty of road classification. The main data sources include aerial and satellite images with different resolution and spectral characteristics, and DTMs sometimes. Some automatic methods use existing GIS data as cues to improve extraction accuracy and reliability or to detect changes for updating (Zhang et al., 2001). Contextual information is taken into account to guide the extraction of roads (Ruskone, 1996). Much of the work is concentrated on low resolution (a few pixels in width), primarily rural areas rather than urban areas due to the complexity of urban scenes (Price, 1999; Hinz and Baumgartner, 2000).

As has been demonstrated in prior research, the reflectance or spectral response as the only resource for road extraction is not sufficient, especially in urban areas. The limiting factors are reflectance depending on viewing conditions and illumination, variability of the surface materials occlusion and resolution (Price, 1999). The similarity in reflectance between objects

leads to a high rate of misclassification. For example, in aerial and high-resolution satellite imagery, the roads are hardly distinguishable clearly from buildings when they are coated with same materials and thus similar gray patterns present. The situation becomes worse when roofs or walls of tall buildings occlude or cast a shadow on the streets in built-up urban areas where neither automatic nor semi-automatic road detection methods could work reliably. In addition, most existing approaches work in 2-D images, thus neglecting valuable information inherent in 3-D processing when using elevation data.

A successful strategy should use specific techniques and fuse most fruitful data to handle different tasks with different road types and contextual complexity in the scene. Using a DSM, the occluded or shadowed areas aforementioned can be derived with the sun position known. This will provide useful information to explain and then bridge the gaps between broken road segments. Zhang et al. (2001) used height information derived by subtracting the DTM from the DSM to reason if a region is on the ground and to compensate the missing information in classification of aerial color images. With increasing availability of lidar data, exploiting lidar elevation information and imagery for road extraction has also been initially investigated. The lidar data is found to be able to improve the analysis of high-resolution image data for detecting buildings and roads especially in urban areas. In cases when shadows cover objects of interest, their shape can be well described due to height information (Hofmann, 2001).

Lidar intensity data has good separability if the wavelength of the laser used is suitable for ground materials. Song et al. (2002) calculated a transformed divergence measure for intensity data to compare relative separabilities between object classes including asphalt road, grass, building roof and tree. In particular, the separabilities for asphalt road vs. grass and asphalt vs. tree are rather high. By fusing elevation information, the intensity data is a valuable data source for road classification (Hu and Tao, 2003). The use of multi-return range data can ease the deletion of penetrable vegetation. Alharthy and Bethel (2003) developed a similar algorithm to detect roads in urban areas using the intensity data and range data. However, the intensity data is usually very noisy due to the continuously varying reflection angles. Some materials appear with different reflectivity as the inclination angle and thus the reflection angle changes. To better represent the material's reflectance characteristics, the intensity values have

to be normalized by the laser reflection angles (Song et al., 2002). However, this is not practical for end users because the normalization requires knowing the orientation information at the time of acquisition.

Another major factor influencing road extraction is the requirement of interaction between the algorithm and an operator. The semi-automatic strategy requires a human operator to input seed points interactively and optional information such as road width etc. to guide the extraction procedure. The optimal paths between seed points are then found by profile or template matching (Airault et al., 1996; Vosselman and de Gunst, 1997; Hu and Tao, 2003a), multi-resolution approach (Couloigner and Ranchin, 2000), Snakes or energy minimizing approaches (Laptev, 1997; Gruen and Li, 1997), dynamic programming (Fischler et al., 1981; Guen and Li, 1994; Mertlet and Zerubia, 1996), cooperative algorithms (McKeown and Denlinger, 1988), or Kalman filtering (Vosselman and de Knecht, 1995). The Snakes or energy minimizing approaches work by defining appropriate energy functions based on radiometric and geometric assumptions for features of interest. They refine the solution to minimize the energy function beginning with an initial estimate of the feature. To extract meaningful features successfully, template matching technique requires that the features possess similar gray patterns. Additional geometry constraints are often utilized to guide the matching. If more than one image are used, this can be done in 3-D (Gruen and Li, 1997).

Great effort has been paid for increasing the automation of road extraction. The automatic strategy finds candidate road segments by edge or line detection methods, and then tracks, verifies and links accepted segments by perceptional organization methods (Wang and Trinder, 2000; Hu and Tao, 2003b) or knowledge based methods (Stilla and Hajdu, 1994; Trinder et al., 1997) to form a complete road network. Although the full automation is likely not reliable in many cases, it is the ultimate goal. Some semi-automatic approaches can be extended to fully automatic ones by means of automatic seed point detection (Zlotnick and Carnine, 1993; Mayer et al., 1997).

The roads extracted automatically may have a low qualified representation at various aspects, and should be refined. In Vestri and Devernay (2001), all the junctions are

processed at the same time. For each type of junction, they randomly sample two points in the different braches, estimate the position of the junction and compute residuals for all random sets. Then the solution that minimizes the median of residuals is selected. Wiedemann (2002) presented an approach for improving the extracted road crossings as well as a scheme for the quality evaluation of the results. The topology reconstruction detects and removes all cycles within the crossing areas. The geometry reconstruction examines all combinations of main and branching roads, which are possible according to the given road segments and typical road crossing models. The combination of main and branching roads that reaches the best score is selected as the final road crossing.

Many road models have been developed utilizing some basic road characteristics. Road characteristics can be classified into five groups: geometric, radiometric, topological, functional and contextual properties (Garnesson et al., 1990). Example characteristics are large length, small width variance, directional consistency, symmetry, homogeneous gray levels, good contrasts with both sides, intersecting each other, and so on (Vosselman and de Knecht, 1995; Barzohar and Cooper, 1996; Fischler and Heller, 1998). Different combinations of the above characteristics should be used to model roads in different resolution levels and contexts (Baumgartner et al., 1997; Mayer et al., 1997; Price, 1999; Wang and Trinder, 2000). Hinz and Baumgartner's (2000) strategy is based on a detailed road model including lanes, road markings and their context using both image and DSM information so that the inherently high complexity of urban scenes is reduced.

In high-resolution imagery, a road is often modeled as a continuous and homogeneous ribbon, which has internal structures such as lane markings and roadsides. Existing experiences show that the use of road models and varying strategies for different types of scenes are promising. The fusion of centerlines from low resolution and roadside edges from high resolution (<1 m) has proven to be very helpful in inferring more reliable results (Steger et al., 1995; Trinder and Wang, 1998). In Couloigner and Ranchin (2000), a novel multi-resolution approach is developed for semi-automatic extraction of streets. The accurate positions of roadsides and central reservations are determined through the analysis of multi-level wavelet coefficients.

Instead of utilizing the above characteristics describing individual road segments, local or global geometric and topological constraints can discover grid layouts of road networks. Price (1999) designed a feature-based hypothesis and verification paradigm to find urban street grids from single or multiple aerial images by applying a local grid constraint. The method assumes that roads have visible edges without significant occlusions. An initial seed intersection that provides the size and orientation of the regular grid is selected manually. Road crossings are junctions of individual road segments of approximately constant width and height. Then the grid is iteratively expanded by adding new units. "In each iteration, the new segments are refined and evaluated by simultaneously matching their sides to image edges. Thus longer portions of the roadsides must be visible at least in one of different overlapping images" (Price, 1999). During final verification, height information and contextual knowledge are used to adjust the positions of several consecutive road segments and to remove short portions. By applying a global grid constraint, Hu and Tao (2003) proposed a procedure to reconstruct the street grids of the 3-D grid road network in urban areas from lidar data also utilizing a road segment based hypothesis and verification paradigm.

2.4 ALGORITHMS FOR BUILDING EXTRACTION

Buildings are the most important artificial objects consisting of piece-wise and continuous surface segments over the ground. Different data sources have been used for the building extraction task. These data sources include aerial and satellite images, 2-D building ground plans from GIS databases, and high-quality DSMs. The task of building extraction is to determine building locations, ground elevation, orientations, building size, rooftop heights, etc. Most buildings can be described to sufficient details in terms of general polyhedra, i.e., their boundaries can be represented by a set of planar surfaces and straight lines. Further processing such as expressing building footprints as polygons is preferable for storing in GIS databases.

Like most feature extraction tasks, building extraction can be implemented in either semi-automatic or automatic strategies, and data-driven and model-driven techniques are commonly used. Some algorithms process the raw lidar point clouds directly or grid-based images converted from lidar data; others algorithms use these two data structures at different

processing stages. I concentrate on algorithms processing the data acquired by lidar systems.

The semi-automatic building extraction approaches often prepare a set of building primitives for typical house types and roof shapes. Human operators place appropriate primitives and combine them to model complex structures. Various automated methods could assist operators in measuring and refining 3-D wire frame models. Brenner (1999) reviewed several interactive modeling tools for 3-D building reconstruction including ObEx, the so-called Stuttgart approach and CC-Modeller. The former two measure building primitives using several (monoscopic view) aerial images or lidar data and 2-D ground plans. While the latter uses stereo measurement of points in aerial images and uses two steps. In the first step, a structured point cloud containing all eaves and ridge points is obtained using strictly manual point measurement. The second step is automatic and consists of grouping points into planar faces and the generation of roof and wall faces.

In Rottensteiner (2001), a method for semi-automatic building extraction together with a concept for storing building models alongside with terrain and other topographic data in a topographical information system is presented. His approach is based on the integration of building parameter estimation into the photogrammetric process applying a hybrid modeling scheme. A building is decomposed into a set of simple primitives that are reconstructed individually and are then combined by Boolean operators. The internal data structure of both the primitives and the compound building models is based on the boundary representation methods.

The automatic building extraction methods can be fulfilled by two sub-procedures, i.e., building detection and building reconstruction (Weidner, 1995), which may not be clearly distinguishable. Full automation of building reconstruction is not yet reliable enough for practical production in most cases due to the great complexity of building architecture. Automatic extraction of buildings would go a long way to making more wide use of available geo-spatial data sources possible such as ground plans.

Some researchers have focused on the automatic extraction solely or mainly based on

DSMs. The methods using only lidar data mainly utilize the geometric properties of buildings, and have to go through the building detection and reconstruction steps. In Weidner and Forstner (1995), the DSM is computed using image and feature pyramids and the final surface is then refined by means of local adaptive regularization techniques. Building detection is based on the fact that buildings are higher than the topographic surface, which is estimated using mathematical morphology on the DSM. The window size requires a priori knowledge of the maximum building size in the scene. Buildings are reconstructed depending on their complexity. Two kinds of parametric models are used for simple buildings with either a flat or a symmetric sloped roof. Prismatic models are used for complex or connected buildings.

In Maas and Vosselman (1999), two techniques for the determination of building models are developed. Based on the analysis of invariant moments of point clouds, closed solutions for the parameters of a standard gable roof building model with a rectangular ground plan are derived from 0^{th}, 1^{st} and 2^{nd} order moments. Asymmetric deviations like dorms on roofs can be modeled too. Inhomogeneity in the point distribution will lead to biased parameters. Models of more complex buildings are determined using a data driven technique based on intersecting planes that are fitted to triangulated point clouds. Most problems occur in the determination of the outline of the building, especially when trees are near the building. Vosselman (1999) presented an approach based on the detection and outlining of planar faces. The planes of the faces are determined by clustering points. The outlines are determined by a connected component analysis assuming that all the edges are either parallel or perpendicular to the main building orientation.

Building surfaces including roofs and walls can be roughly approximated by constructing a TIN for points composing a building. The TINs of 3-D points sampled on object surfaces and the simplification and refinement methods have been extensively studied for approximating object surfaces (Heckbert and Garland, 1997). The simplification method is a fine-to-coarse approach, and starts with an exact fit, and creates approximations with less and less details (Wang and Schenk, 2000); while the refinement method is a coarse-to-fine approach and starts with a minimal approximation, and generates more and more

accurate ones (Hu et al., 2003). Most existing roof reconstruction methods are based on the simplification concept. They first try to aggregate the points that possibly belong to separate patches of a complex roof, and a plane fitting is then performed to get parameter values for each planar patch. The plane detection methods reported in literature include clustering of triangles, 3-D Hough transformation, and clustering of 3-D points with or without using ground plans (Maas and Vosselman, 1999; Vosselman and Dijkman, 2001; Gamba and Houshmand, 2002). Those adjacent triangles with close normal directions may be grouped, and a plane equation is fitted to the vertices of the grouped triangles. Then the fine wire-frame model of a building's surface can be obtained by calculating and organizing the intersection lines between planes.

Wang and Schenk (2000) proposed an edge-based building detection and a TIN-based building reconstruction. The building models are reconstructed by triangulating each cluster of identified building points and grouping those fragmentary triangles into piece-wise planes. Finally, the tri-intersections of those average planes are used to derive building corners and their relative orientations. The study area has nine buildings ranging from large to median to small sizes (Wang and Schenk, 2000). All the buildings in the area have rectangular or near rectangular shape with peaked roof, flat roof and multiple level flat roof, respectively. The range data play a major role in the building extraction although it may be not enough to complete the processing in many cases using only range data. The combination of geometric measures is proven effective to classify buildings.

The 3-D Hough transformation is used to extract planar faces in Vosselman and Dijkman (2001). However, plane extraction with 3-D HT has a high computational complexity determined by the number of 3-D points and the size of a discretization of the two angles. In Elaksher and Bethel (2002), the parameter space of the 3-D HT is reduced from three to two by using a simplified model assuming that only one of the roof slopes is not zero along or perpendicular to the main building orientation.

Some methods focus on the building detection stage. The derived building footprints or polygons can be used to update the building basemap that is an integral part in spatial

databases of GIS. Baltsavias et al. (1995) used an edge operator, mathematical morphology, and height bins for the detection of objects higher than the surrounding topographic surfaces. Hug (1997) showed the detection and segmentation of houses from ScaLARS height and intensity data based on morphological filtering with successive progressive local histogram analysis; in addition, they use the laser reflectivity measure for discerning manmade objects from vegetation via binary classifications. In Hu et al. (2003), several algorithms, including the constrained searching in Hough space, enhanced Hough transformation and sequential linking technique, are developed to reliably express building footprints as rectangles, quadrangles or polygons. These algorithms utilize the hypothesis verification paradigm and a final refinement.

The active contour model (see Section B.1.11) is used to locate the boundary of a building in Yoon et al. (1999). Hu and Tao (2002) also tested the active contour model to locate building boundaries using greedy algorithm, and found several drawbacks, including the requirement of initial boundary position, possible shift to neighboring buildings during iterations and the difficulty of reconstructing regular shapes using a priori knowledge (e.g., orthogonality). Research effort has been made to rectify building boundaries detected automatically. Sester (2000) presented solutions for generalization problems using least-squares adjustment, focusing on the scale dependent representations. In Vestri and Devernay (2001), angle constraints are applied to refine the corners and junctions of polygon models, and the problem is solved by optimising an objective function to preserve the global consistency.

An important task in building detection is to distinguish between buildings and vegetation. The discrimination between buildings and vegetation based on a Bayesian nets classification algorithm using local geometric properties is discussed in Brunn and Weidner (1998). Zhan et al. (2002a) applied an object-based classification to detect building footprints. First, the image segments belonging to the building class are identified by the vertical wall analysis, which is to examine the sliced lidar DSMs; then colour infrared imagery is used to calculate the NDVI, which is then assimilated to refine the identified building segments (Zhan et al., 2002b).

Rottensteiner (2001) found that roof structures could be extracted as a first step toward the reconstruction of polyhedral building descriptions. The use of shape measures based on differential geometric properties of range data for building detection is not sufficient and also poses high requirements on the resolution and quality of the data. In this situation, multiple returns and lidar intensity data could help much. Elberink and Maas (2000) presented their work on segmenting 0.5-m to 1-m lidar data in an unsupervised *k*-means classification of objects (e.g., buildings and trees) using anisotropic height texture measures. The texture is defined quantitatively and qualitatively by height, variation of height in local windows and measures such as contrast and homogeneity. An important aspect of the work is the analysis of the benefit of intensity data as well as first and last returns range data. In Hu et al. (2003), the first and last returns lidar range data are used to produce the VSM that represents penetrable high vegetation objects such as trees; then the binary objects in the up segmented DNM that have non-empty intersections with the VSM are deleted with only buildings retained.

The methods using both DSMs and auxiliary data such as GIS maps and multi-spectral images could overcome the drawbacks of specific imaging sensors. The integration of lidar data and ground plans are shown to be successful, and detailed reconstruction of buildings can be obtained automatically even for lidar data with relatively low point densities. This type of approach has the advantage of skipping the building detection step with ground plans known and focusing on building reconstruction stage. Henricson et al. (1996) used information from CIR images to separate elevation blobs detected in a DSM from stereo image matching into the classes of buildings and trees. Lemmens et al. (1997) showed the fusion of lidar altimeter data with 2-D digital maps in a topographical database to derive heights for roofless cube type building primitives.

Haala and Anders (1997) demonstrated two approaches aiming on the combination of DSMs, aerial images and ground plans for the reconstruction of 3-D buildings.
The ground plans provide very precise information about the building outlines. The first approach extracts breaklines from both DSMs and image data. Then the breaklines of

high reliability are combined with the gray value edges of high geometric accuracy to reconstruct a rather simple type of buildings. The second approach extracts planar surfaces likely to be roof planes from the DSMs and uses polyhedron as the building model to reconstruct very general type of buildings. It utilizes given ground plans as a priori information and extracts planar surfaces likely to be roof planes from DSMs.

Haala et al. (1998) derived parameters for 3-D CAD models of basic building primitives by least-squares adjustment minimizing the distance between a laser DSM and corresponding points on a building primitive. The building boundaries were derived from ground plans. The implementation was limited to four standard building primitives and their combinations. Further refinement has to be performed interactively. Haala and Brenner (1999) combined multi-spectral information provided by an aerial CIR image with geometric information from a lidar DSM. A pixel-based classification is applied for the extraction of buildings, trees and grass-covered areas, whereby the normalized DSM is used as an additional channel in combination with three spectral bands. Additional constraints are obtained for reconstruction by using the assumption that the given ground plan is correct and exactly defines the borders of the roof.

Brenner (2000) also presented the reconstruction of building of complex structures using lidar DSMs and ground plans. Roof surface primitives are segmented, and a rule-based approach decides which segments can be explained by the chosen building model. Finally, the roof is built from the primitives that have been accepted, closing gaps that are caused by the deletion of unexplainable regions. In Ameri (2000), building models are firstly created based on DSMs, and are then verified by back-projecting them to images. Matching the model edges with image edges, the accuracy of the model parameters can be increased especially with respect to the building outlines.

The quality evaluation for building detection results has been initially addressed. Geibel and Stilla (2000) compared four segmentation procedures using lidar data with a density of four points/m^2, and proposed an evaluation function to estimate the segmentation quality of a complete scene. The evaluation function subjectively weights measures for

the over-segmentation and under-segmentation. However, at present there is a lack of objective measures concerning reconstruction times, success rates and reconstruction quality, especial when a human operator is the part of a semiautomatic system and different ancillary data are used (Brenner, 1999; Rottensteiner, 2001; Lemmens et al., 1997; Vosselman and Dijkman, 2001).

2.5 SUMMARY

Wide-area lidar data is rapidly becoming the standard data source for many mapping applications. Research effort has been made to investigate areas where lidar technology may provide significant advantages such as allowing value-added products to be generated and offering great cost reduction over traditional methods. In this section, I introduced various representative algorithms developed for analysing lidar data and possibly other remote sensing data for extraction of DTMs, roads and buildings. Automated feature extraction would go a long way to making more wide use of geo-spatial data sources available.

The availability of a suite of COTS software tools will eventually replace the proprietary solutions common today. This will open up the processing workflow and change the existing value chain, and thus shift the primary product from lidar point clouds to the more basic geo-referenced features. Lidar technology will certainly continue to proceed technically and to find its new applications. The potential integration with other imaging sensors is expected to put lidar technology on a new level in terms of system performance and mapping capabilities.

CHAPTER THREE

DIGITAL TERRAIN MODEL GENERATION

The foundation of most mapping products are DTMs, which are required for many geospatial applications. In this chapter, a hierarchical terrain recovery algorithm is developed for DTM generation using lidar data. The method for processing single-return range data is described in Sections 3.2 to 3.4. In Section 3.5, the above methodology is modified to utilize more information contained in multi-return range and intensity data. In Section 3.6, I discuss the quality of testing results for several lidar datasets, and assess the performance of the algorithm.

3.1 INTRODUCTION

Highly detailed and accurate terrain data is one of the most critical components of many mapping, engineering and natural resource management projects. The demand for high-quality DTMs is increasing significantly as the GIS community advances toward 3-D technology and virtual-reality environments. Many geospatial applications, such as urban planning, landscape analysis, transportation and hydrological watershed analysis, need to use DTMs. The primary remote sensing data for DTM production include aerial and satellite images, IfSAR data, lidar data, and GIS data. To generate a DTM from lidar data, one has to remove points falling on aboveground features, and to interpolate between the remaining terrain points falling on the bare Earth surface.

As reviewed in Sections 1.3 and 2.2, many algorithms have been developed to generate DTMs using lidar data in the last decade [4]. Related techniques have been extensively researched for exploiting terrain information contained in lidar data. I can say that a reliable and efficient algorithm for DTM generation from wide-area lidar data should take following empirical rules into account to process complex scenes.

[4] I may include a minimum repetition necessary for further analysis.

- The local lowest points (excluding blunders) in a moving window of large size are most reliable terrain points (Weidner and Forstner, 1995; Kilian et al., 1996; Hug and Wehr, 1997; Petzold et al., 1999; Axelsson, 2000; Masaharu and Ohtsubo, 2002; Wang et al., 2001). The window size has to be large enough so that there must exist at least one point hit on the topographic surface in any window. The terrain surface may be approximated by an interpolation of these terrain points, and obviously, has lower elevations compared with the ground truth on average.
- The topographic (e.g., ridge, slope and valley) points in filtering windows of varying sizes are preferable terrain points, which are identified by estimating surface curvature based on the local geomorphology. These terrain points will retain more details of the terrain surface (Fraser and Jonas, 2001; Wang et al., 2001; Tao et al., 2001; Schickler and Thorpe, 2001).
- The natural terrain surface could be approximated using piece-wise smooth facets (Haugerud and Harding, 2001). This smooth condition is used to identify non-terrain points based on terrain points utilizing terrain slope and/or relief information (Vosselman, 2000; Fraser and Jonas, 2001; Sithole, 2001). The use of extra data such as breaklines could help to identify terrain points at a larger scale and thus avoid Type I error at places where geomorphological structures occur (Petzold et al., 1999; Schickler and Thorpe, 2001; Sithole and Vosselman, 2003).
- The coarse surface obtained by interpolating terrain points with insufficient density is under-estimated, thus iterations are necessary (Hug and Wehr, 1997; Kraus and Pfeifer, 1998; Axelsson, 2000; Fraser and Jonas, 2001; Wang et al., 2001). During iterations, the coarse surface will be refined by subsequent introduction of denser terrain points by comparing information between the finer data and the coarse surface and also by the recursively refined knowledge about the ground coverage, which allows for improving the estimation of parameter thresholds. The data structure suitable for this multi-resolution and terrain adaptive strategy is a pyramid (Tao et al., 2001; Wack and Wimmer, 2002).

The first rule is a common idea for all of existing algorithms, and the latter three are supported partly by them. But none of these algorithms support all the four rules due to

different goals and strategies. I address the DTM generation task from two aspects:

- *Scene complexity*. The bare Earth is composed of smoothly or abruptly changing terrain surfaces, and is partly covered by low- to high- density natural and cultural features. Different terrain and coverage types in a scene lead to the difficulty in recovering the bare Earth from sampled lidar points. Existing DTM generation algorithms are not yet competent for practical industry production, and may fail in certain areas due to the complexity of scenes with varying terrain and coverage types.
  - These algorithms perform well in scenes of low complexity (Sithole and Vosselman, 2003), but extensive manual editing is required to obtain a clean representation of the ground surface in complex scenes (Tao and Hu, 2001).
  - These algorithms are intended to process certain scenes with expected terrain and/or coverage types. Algorithms designed to process forested areas likely perform poor when processing urban areas, or vice visa. Thus different versions of these algorithms have to be customized for processing different situations. For example, Kraus and Pfeifer's (1998) filtering method is originally designed to process forested areas, and has been modified to process urban areas in Schickler and Thorpe (2001) and Pfeifer et al., (2001). I will develop the capability of processing complex scenes in a single workflow.
  - Some algorithms have a low efficiency in their workflows to process slightly complex scenes with terrain and coverage variations. In Wang et al. (2001), the basic algorithm is suited to process urban areas due to the use of fixed thresholds in identifying non-terrain points. Then a self-diagnosis procedure is designed to be able to process both urban and mountainous areas in one scene. It applies the urban-oriented algorithm one more time to detect the hills removed in the first pass and one more time again to recover them. To conquer this drawback, I will determine the thresholds assimilating both terrain relief and slope information.
  - Most algorithms are data driven techniques and are flexible for processing lidar data acquired by ever-improving lidar systems (Tao and Hu, 2001). The model-driven techniques are more limited to scenes of lower complexity since they often make some unrealistic assumptions about contents of the scenes (Elmqvist, 2001).

- *Data fusion*. Current lidar systems provide information about multiple returns of each pulse and the reflectivity properties of ground materials (Optech, 2003). However, exiting DTM generation algorithms focus on processing single-return range data only (Tao and Hu, 2001; Sithole and Vosselman, 2003). Actually, multi-return range data can be used to detect penetrable vegetation (mainly tree) points, which are obviously non-terrain points (Hu and Tao, 2003; Alharthy and Bethel, 2003). Intensity data can be used to classify land-cover classes such as asphalt road vs. grass and asphalt vs. tree (Song et al., 2002), and road points are obviously terrain points. The above information is valuable for reliably identify terrain and non-terrain points, but is omitted by existing algorithms. I will also show that such information has an exceptional use for adaptively adjusting parameter values in my algorithm.

In this chapter, a hierarchical terrain recovery algorithm (HTRA) is developed to build an efficient framework of data-driven DTM generation for single- or multi-return range and intensity data. The HTRA generates DTMs fully automatically for complex scenes by using adaptive and robust filtering and interpolation techniques. It generates range pyramids, identifies topographic points, and interpolates terrain elevations at non-terrain points in a hierarchical approach. The smooth condition improves the processing efficiency by integrating information about terrain relief, slope and data density. When road points are detected using intensity data and VSMs are derived using multi-return range data, the algorithm is able to adaptively adjust parameter values in the smooth condition to be suited to process changing contents in a scene.

3.2 DTM GENERATION FOR SINGLE-RETURN RANGE DATA

I first develop the HTRA to automate DTM generation for single-return range data, which is the dataset available in most cases.

3.2.1 Overview of the Proposed Methodology

The algorithm will identify terrain points by finding local minima and other topographic

points, and recover the terrain surface in a coarse-to-fine manner. First, after screening the blunders, the scattered 3-D points are transformed into a grid-based range image by selecting the point of lowest elevation in each grid. Then, an image pyramid is generated. The top-level image is hypothesized to be a coarse DTM if its grid size is larger than the largest non-terrain object. Finally, the coarse DTM is refined hierarchically from the top level to the bottom level. At each level, denser terrain points are identified, and the non-terrain points are replaced by interpolated elevations using surrounding terrain points. The bottom-level image represents the expected bare Earth surface.

3.2.2 Workflow of DTM Generation

The method can be divided into four steps:

Step1: lidar data pre-processing. The raw lidar data are filtered to discard blunder points, which have too low or too high elevation values, or very large intensity values that do not match their surroundings (see Section 4.2.2). Many blunder points can be discarded promptly if a priori knowledge about the terrain relief of the ROI is available.

Step 2: lidar image processing. The filtered lidar points are transformed to a grid-based range image, in which each grid is assigned the elevation of the lowest point in that grid. Then the raw range image is processed to fill void areas and correct distortions.

Step 3: hierarchical terrain recovery algorithm. This is the core step. First, the raw range image is used to generate an image pyramid, the top-level of which is a coarse DTM. This coarse DTM is refined hierarchically by processing image levels top down in the pyramid. When processing a level, its immediate higher level is treated as the reference surface, based on which the terrain points are identified using the smooth condition, and the elevations at non-terrain points are interpolated from surrounding terrain points.

Step 4: DTM refinement. Post-filtering is performed on the resultant DTM to improve its quality for mapping purposes. This smoothing attempts to correct the influence of speckles and undesired non-terrain undulations of the DTM. At last, the void areas in the

raw range image may be duplicated to preserve water regions.

The quality of the derived DTM is subject to thresholding parameters used in the algorithms. The optimal values of these parameters may vary with the scene complexity. I use a priori knowledge, if available, including the terrain relief range, the maximum and minimum building sizes, heights and areas, the maximal tree height, etc to determine the parameters empirically first, and then adjust them adaptively if multi-return or intensity data is available.

3.3 LIDAR IMAGE PROCESSING

3.3.1 Create Grid Images

The MBR of the ROI is calculated first. Then the ROI is divided into regularly spaced grid cells. To make full use of the data resolution, each grid cell contains one point on average. The grid size is estimated by

$$s_c = \sqrt{A_{ROI} / n_{ROI}} \tag{3.1a}$$

where n_{ROI} is the number of points and A_{ROI} is the covered area. The center of each grid is assigned the elevation or intensity value of the lowest point in that grid. This is equivalent to shifting the selected point to the grid center while preserving its elevation. Range or intensity images in grid format are thus created. The void grids are assigned a special value outside of the possible terrain relief range, that is -9999 for the ESRI null value.

3.3.2 Process Lidar Images

The void areas in the raw range or intensity image are recursively filled up to a certain gap. A void pixel is updated to the minimum of the valid pixels in its neighborhood if more than a half of pixels are valid in that neighborhood. The gap is estimated by

$$s_{gap} = h_{\max} \cdot \tan a \tag{3.1b}$$

where h_{max} is the maximal building height, and a is half of the scan angle (e.g., 16^o). So only small holes with valid data around are filled, and big holes such as large water areas are reserved. Because only one point is selected in each grid cell during the creation of raster images, possible zigzags can occur at places where the elevations change suddenly, especially around building boundaries and corners. These alias effects and random noises are suppressed by applying a multi-stage median filter (see Section B.1.4).

3.3.3 Determine the Largest Window Size

The largest window size should be not less than the smaller dimension of any non-terrain object in the ROI, ensuring that there must exist at least one terrain point in any window. If an object is larger than the largest filtering window, then the inner parts of that object will be wrongly treated as terrain level. Using the largest dimension of all the buildings is not appropriate because this often overestimates the optimum size. I detect edges in the range image and encode the connected edge pixels into individual object contours (see Section B.1.11). The window size in pixels is estimated by

$$s_w = \min(\sqrt{A_o}, P_o / 4) \tag{3.1c}$$

where A_o and P_o are the area and the perimeter, respectively, of the object whose area is largest among all the contours. Here, s_w is also limited to 120 m that is suitable for datasets tested in this thesis.

3.4 HIERARCHICAL TERRAIN RECOVERY ALGORITHM

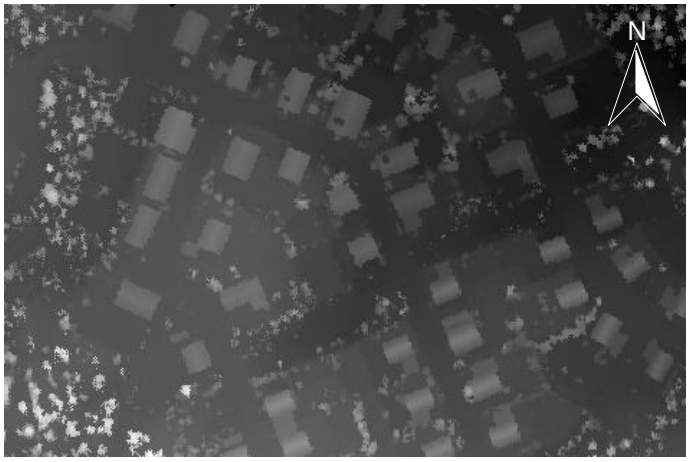
The algorithm generates an image pyramid from the range image, identifies terrain points, and interpolates terrain surfaces at non-terrain regions in a hierarchical approach.

3.4.1 Generation of Image Pyramids

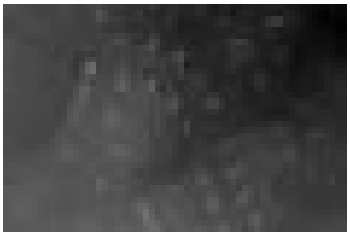
The data structure used to represent image information can be critical to the successful completion of a data processing task. One structure that has gained considerable attention is the pyramid (see Section B.1.2). The number of pyramid levels is determined by

$$n = \lceil \ln s_w / \ln k \rceil + 1 \qquad (3.1d)$$

where *k* is the pyramid scale that is determined by the size of the smallest object. Thus, the actual grid size of the top level s_{top} is equal to k^{n-1} pixels, which is not less than the largest window size s_w. A small pyramid scale will result in more levels, which allow for smoother interpolation with the price of heavier computational burden.



(a) The bottom level, 315 x 474



(b) The intermediate level, 53 x 79 (enlarged 3 times)

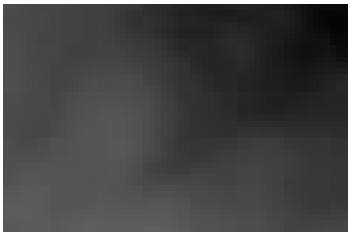
(c) The top level, 9 x 14 (enlarged 9 times)

Figure 3.1. Image pyramid generation (Village site)

The first level (i.e., bottom level) duplicates the processed range image. The different pyramid levels are created by selecting lowest points in image grids. The pyramid generated for the Village data consists of three levels that are shown in Figure 3.1.

Starting from the bottom (i.e., the first) level (see Figure 3.1a), the lowest point in each distinct *k* x *k* window is assigned to the corresponding grid at a higher level until the top (i.e., the n^{th}) level is reached. During the creation of pyramidal images, non-terrain points are removed gradually, and the top-level image (see Figure 3.1c) is a coarse approximation of the bare Earth surface when its grid size is large enough to see on terrain at any place. Obviously, the DTM interpolated from the top-level image will result in a lower terrain surface than the bare Earth surface on average.

(a) The DTM at the top level (enlarged 9 times)



(b) The intermediate level (enlarged 3 times)

(c) The DTM at the bottom level

Figure 3.2. Hierarchical terrain recovery (Village)

3.4.2 Hierarchical Recovery of Terrain Surfaces

The terrain surface is recovered hierarchically from the top level to the bottom level of the pyramid. During the iterations, more terrain points are identified, and the non-terrain points are replaced by interpolated points using those already identified terrain points.

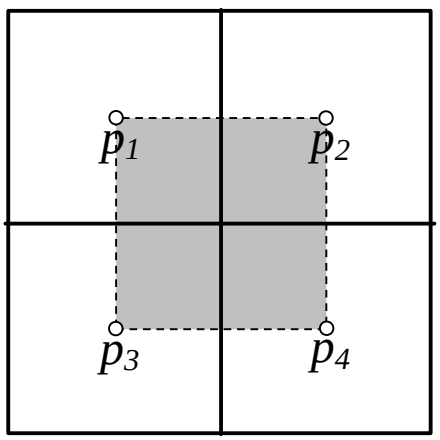
Figure 3.2 shows the three levels during the terrain recovery procedure. The coarse DTM in Figure 3.2a is exactly the same as in Figure 3.1c. The procedure starts from the $(n\text{-}1)^{th}$ level that is called the current level under processing, and its immediate higher level, here the top level, is treated as the reference level. The bare surface can be recovered by comparing elevations between the current level and the coarse surface interpolated from the reference level. The bottom-level image (see Figure 3.2c) is an initial DTM.

In the iteration, a grid cell at the reference level is expanded to a surface patch with k x k grids at the current level. In Figure 3.3a, the four small circles represent the centers of four grids p_u (u=1…4) at the reference level, and Z_u (u=1…4) denote their elevations. These four points are assumed to be on the true terrain surface and are used as reference points at the current level. The shaded square formed by these four points corresponds to a k x k window at current level as shown in Figure 3.3b. This window is mapped to the shaded area in Figure 3.3c or Figure 3.3d depending on the parity of k.

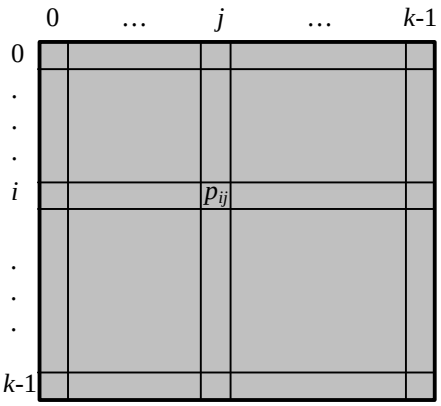
To obtain an approximation to the terrain surface at a higher resolution, I assume that a simple facet may fit the 2.5-D surface patch within the current k x k window. To be able to interpolate grids around the boundary of the current level, I extend and duplicate two rows (columns) outside the first and last rows (columns) at each reference level. Let p_{ij} ($0 \le i, j \le k-1$) denote a point at position (i, j), which is interpreted as integer row and column coordinates relative to the top-left corner of the current window as shown in Figure 3.3b. The elevation at p_{ij} is estimated from the reference points by bilinear interpolation using Equation 3.2a if k is even or Equation 3.2b if k is odd.

$$\hat{Z}_{ij} = \frac{1}{k^2}\big((k-i-0.5)\cdot(k-j-0.5)\cdot Z_1 + (k-i-0.5)\cdot(j+0.5)\cdot Z_2 + (i+0.5)\cdot(k-j-0.5)\cdot Z_3 + (i+0.5)\cdot(j+0.5)\cdot Z_4\big) \tag{3.2a}$$

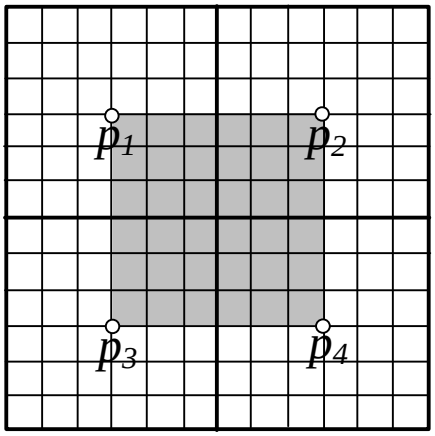
$$\hat{Z}_{ij} = \frac{1}{k^2}\big((k-i)\cdot(k-j)\cdot Z_1 + (k-i)\cdot j\cdot Z_2 + i\cdot(k-j)\cdot Z_3 + i\cdot j\cdot Z_4\big) \tag{3.2b}$$



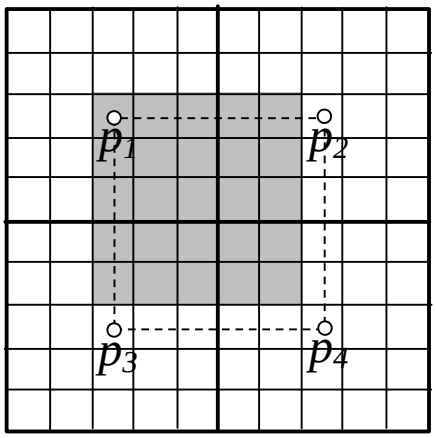
(a) Four grids at the reference level



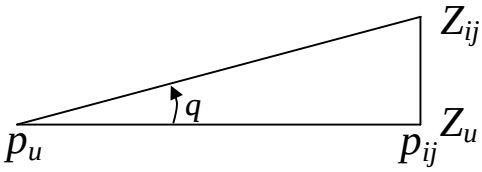
(b) *k* x *k* window at the current level



(c) The current window (*k* is even)



(d) The current window (*k* is odd)



(e) The profile connecting p_{ij} and p_u

Figure 3.3. Structure of the *k* x *k* window

If the current window contains a topographic patch, then the elevations are assumed to change smoothly and no further processing, such as the interpolation described in Section 3.4.2.3, is needed. The premise is that a smooth condition has been established to validate that the elevation at the current level and the estimated elevation from the reference level are reasonably close at every point of that patch. If any point in the current window violates the smooth condition, then that point is classified as a non-terrain point. The

elevation of a non-terrain point should be replaced by a value derived from surrounding terrain points in that window. However, it is hard to estimate a good elevation, which is consistent with the terrain in vicinity, solely using a few terrain points such as the reference points. To recover the terrain details at non-terrain points, topographic points on that surface patch have to be identified to prepare for the interpolation.

3.4.2.1 Establishment of the Smooth Condition

Let Z_{ij} denote the elevation of the point p_{ij} at the current level. The smooth condition compares Z_{ij} with respect to terrain points at the reference level. That is, the difference between Z_{ij} and $\hat{Z}_{ij}$ (Equation 3.2a or 3.2b) at any terrain point should be not larger than a certain threshold as given by

$$\Delta Z_{ij} = Z_{ij} - \hat{Z}_{ij} \le Th \tag{3.3}$$

where Th is the height threshold that is vital for reducing both types of misclassifications. The height threshold takes both local terrain relief and slope information into account, and is defined by Equation 3.4, where ΔR is the range resolution defining the relative accuracy of lidar points in Z direction (e.g., 5 cm, see Table A.1).

$$Th = \min(Th_1, Th_2) + \Delta R \tag{3.4}$$

with

$$Th_1 = c_1 \cdot h \cdot \ln(u+1) / n \tag{3.5a}$$

$$Th_2 = c_2 \cdot \min(1+ | \tan q |, c_3) \cdot s_c \tag{3.5b}$$

The smooth condition utilizes relief information as defined in Equation 3.5a, where c_1 is a parameter default to 1.2, which is an empirical value obtained based on multiple testing results; h is the extreme elevation difference among all the points in the current k x k

window; u ($1 \le u \le n$) is the pyramid level number. The rational behind Equation 3.5a is that local lowest points in windows of varying sizes are more possible to be on the bare surface and local highest points are likely reflected from aboveground features. This relief threshold adaptively changes to be suitable for different resolutions at different pyramid levels. It gives a larger threshold for a lower resolution since non-terrain points have been largely eliminated in that situation.

The smooth condition also considers slope information as defined in Equation 3.5b, where c_2 and c_3 are parameters default to 1 and 3, respectively, also based on multiple testing results; q is the slope angle at p_{ij} relative to the local topography in the current window. The slope angle q is selected among four (or three at the left-top point p_{11} for an odd pyramid scale k) components, each of which calculates a slope of p_{ij} relative to a reference point as shown in Figure 3.3e. Similarly, the computation of q also has two forms relying on the parity of k as given in Equation 3.6a for an even k and Equation 3.6b for an odd k. The rational behind Equation 3.5b is that the terrain elevations usually change following the overall geomorphologic trend in a piece of Earth surface. This slope threshold adaptively changes according to the density of the raw dataset. It calculates a larger threshold when the Earth's surface has been sampled more coarsely and thus more details about non-terrain objects are already lost.

$$\tan q = \frac{1}{s_c} \min\left(\frac{|Z_{ij} - Z_1|}{\sqrt{(i+0.5)^2 + (j+0.5)^2}}, \frac{|Z_{ij} - Z_2|}{\sqrt{(i+0.5)^2 + (k-j-0.5)^2}}, \frac{|Z_{ij} - Z_3|}{\sqrt{(k-i-0.5)^2 + (j+0.5)^2}}, \frac{|Z_{ij} - Z_4|}{\sqrt{(k-i-0.5)^2 + (k-j-0.5)^2}} \right) \tag{3.6a}$$

$$\tan q = \frac{1}{s_c} \min\left(\frac{|Z_{ij} - Z_1|}{\sqrt{i^2 + j^2}}, \frac{|Z_{ij} - Z_2|}{\sqrt{i^2 + (k-j)^2}}, \frac{|Z_{ij} - Z_3|}{\sqrt{(k-i)^2 + j^2}}, \frac{|Z_{ij} - Z_4|}{\sqrt{(k-i)^2 + (k-j)^2}} \right) \tag{3.6b}$$

The smooth condition can be suited to adaptively process most situations occurred in typical lidar datasets. The combination of both relief and slope thresholds by a minimization serves the purpose of minimizing Type II error. Generally speaking, the relief threshold Th_1 is usually selected when processing flat and mountainous areas; while the slope threshold Th_2 is likely selected when processing building roof points and tree points. Therefore, flat grounds as well as hills with smoothly changing slopes will be kept in the DTM correctly all the time.

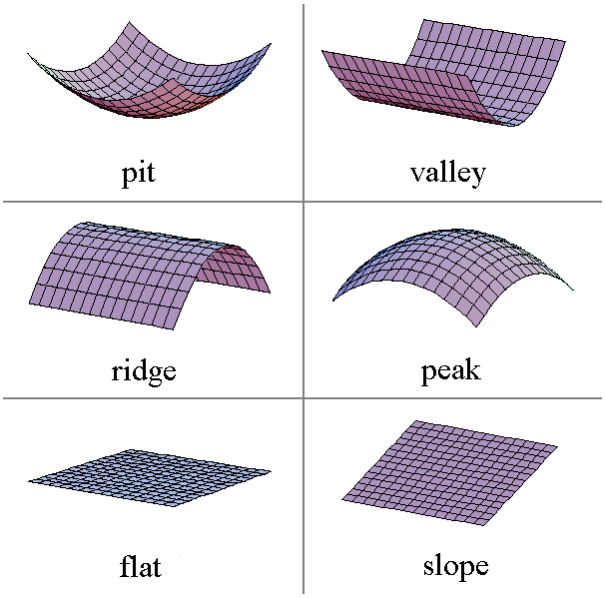


Figure 3.4. Six types of topographic points

3.4.2.2 Identification of Topographic Points

Besides local lowest points, I have to identify other topographic points, which may be sited on the slope, ridge, valley or flat surface. To save computational time, I directly use digital differences to identify six types of topographic points (see Figure 3.4, Li, 1999) without fitting the set of piece-wise smooth functions used by Besl and Jain (1988, see Section B.2.1). It is should be noted that only points satisfying the smooth condition are hypothesized to candidate topographic points, which are then verified by checking the surface geometry formulated in Equations 3.7a to 3.7c, where Z_x' and Z_x'' are the first- and second-order elevation differences in *x* direction, respectively; Z_y' and Z_y'' are the first- and second-order elevation differences in *y* direction, respectively.

- *Pit and valley conditions*.

$$Z_x' = Z_y' = 0 \wedge \left((Z_x'' > 0 \wedge Z_y'' > 0) \vee (Z_x'' = 0 \wedge Z_y'' > 0) \vee (Z_x'' > 0 \wedge Z_y'' = 0) \right) \quad (3.7a)$$

- *Ridge and peak conditions*.

$$Z_x' = Z_y' = 0 \wedge \left((Z_x'' < 0 \wedge Z_y'' < 0) \vee (Z_x'' = 0 \wedge Z_y'' < 0) \vee (Z_x'' < 0 \wedge Z_y'' = 0) \right) \quad (3.7b)$$

- *Flat and slope conditions*.

$$Z_x'' = 0 \wedge Z_y'' = 0 \quad (3.7c)$$

The pit and valley points have the highest priority. The equations describing above conditions are not rigorous in mathematics (Shen et al., 1992; Li, 1999), but are good enough for practical applications. The topographic points identified for Village site are evenly distributed over the whole scene as shown in Figure 3.5.

In Equations 3.7a to 3.7c, the first-order derivatives are calculated in 3 x 3 windows using 1-D Prewitt operators, and the second-order derivatives are calculated using 1-D Laplacian operators (Pratt, 1991). The signs of the elevation differences are determined using a positive zero threshold e as given by

$$d \begin{cases} > 0 & if \;\; d > e \\ = 0 & if \;\; |d| \leq e \\ < 0 & if \;\; d < -e \end{cases} \quad (3.8a)$$

with

$$e = c_4 \Delta R \quad (3.8b)$$

where the parameter c_4 is default to 1 for first-order differences and $\sqrt{2}$ for second-order differences, respectively, since ΔR represents the range resolution (see Table A.1).

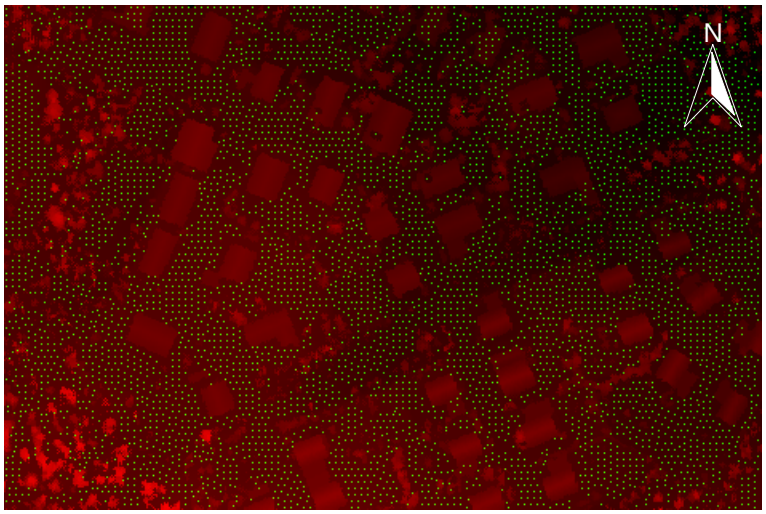


Figure 3.5. Topographic points overlaid on range image

3.4.2.3 Interpolation at Non-terrain Points

If only a few (six or less) terrain points can be identified in the current window, I fit a plane equation to these topographic points and, if necessary, the reference points. This situation often occurs when the current window is located on the building roof or at the boundary of a large building. The plane is given by Equation 3.9, where *i* and *j* are point coordinates; b_u (u=0…2) are coefficients solved by linear regression (see Section B.2.5).

$$Z = b_0 + b_1 \cdot i + b_2 \cdot j \tag{3.9}$$

If more than six terrain points are identified in the current widow, a second-order polynomial facet is used to fit these terrain points to estimate the elevation at a non-terrain point. Compared to a plane equation, the second-order polynomials may keep the fidelity of pits, ridges and valleys especially in mountainous areas. The bivariate polynomial is given by

$$Z = a_0 + a_1 \cdot i + a_2 \cdot j + a_3 \cdot i^2 + a_4 \cdot i \cdot j + a_5 \cdot j^2 \tag{3.10}$$

where i and j are point coordinates; a_u (u=0…5) are coefficients also solved by the least-squares method with equal weighting. The error equations in matrix form is given by

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} = \begin{bmatrix} 1 & i_1 & j_1 & i_1^2 & i_1 j_1 & j_1^2 \\ 1 & i_2 & j_2 & i_2^2 & i_2 j_2 & j_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & i_m & j_m & i_m^2 & i_m j_m & j_m^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} - \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_m \end{bmatrix} \tag{3.11}$$

where m is the number of points; r_u and Z_u (u=1…m) are residuals and elevations, respectively, at the terrain points.

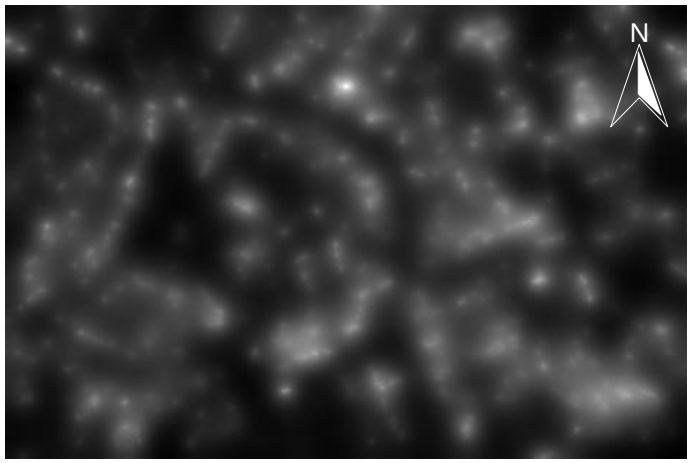
3.4.3 Refinement of DTMs

In the initial DTM, Type I error will occur if hill peaks are wrongly cut or pits are filled due to interpolation using their surrounding points. The HTRA can recover most terrain details since the smooth condition has incorporated geomorphologic information to allow for identification of terrain points on hill slopes and non-terrain points on the aboveground objects. In this section, I locate over-estimated areas where the elevations in the DTM are higher than those in the DSM. This is to detect negative values in the digital non-terrain model (DNM or DNTM) [5]. The DNM refers to the height representation of non-terrain objects, including vegetation, buildings and other objects that stand upon the bare Earth surface. The DNM represents all these aboveground objects upon a flat reference plane, and is the subtraction between the lidar DSM and the DTM. In this sense, the DNM is also called height data in this thesis.

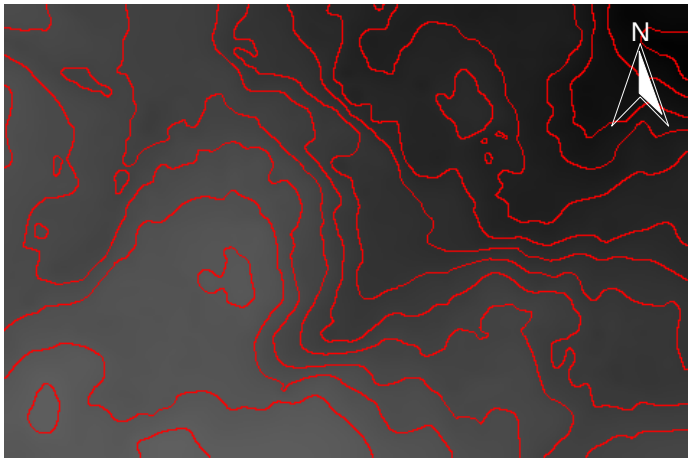
All the positive values in the DNM are reset to zero first, and the sign of negative values are then reserved to obtain an error image that represents the over-interpolated areas in the DTM. The error image is smoothed using moving planes. The plane coefficients are

[5] The DNM is called normalized DSM in Weidner and Forstner (1995).

solved using points within 7 x 7 windows by least-squares adjustment. Once the plane equation is fitted, the elevation of the center point is calculated, and is simply b_0 if the origin has been moved to the center point before fitting. Finally, the DTM is corrected by subtracting the smoothed error image. Figure 3.6a shows the smoothed over-interpolation error image for Village site, which has the maximal elevation difference of 32 cm.



(a) The smoothed over-interpolation error image



(b) The final DTM with 1-m contours

Figure 3.6. DTM refinement (Village site)

Since the DTM is recovered patch by patch at each level, there may exist inconsistencies at patch boundaries. To smooth the DTM, a plane (Equation 3.9) is fitted in each 5 x 5 window excluding the center point by least-squares adjustment. The RMS error for the fitted plane is denoted as $\hat{s}_Z$. To minimize the introduction of Type II error, the points with large positive residuals as given by

$$Z - \hat{Z} > 2.5\hat{s}_Z \quad (3.12)$$

are discarded as non-terrain points, and the plane is fitted again using remaining points. Let Z_c denote the elevation of the center point and $\hat{Z}_c$ its estimated elevation. Then Z_c is replaced by $\hat{Z}_c$ if their difference is larger than a pre-selected threshold. This condition is given by Equation 3.13, where s^2_{DTM} is the noise variance of the DTM, and is equal to the mean squared difference between the DTM and its median filtered version.

$$| Z_c - \hat{Z}_c | > 2s_{DTM} \quad (3.13)$$

Figure 3.6b shows the refined DTM for Village site. Compared with the DSM shown in Figure 3.1a, the non-terrain objects including buildings and trees are successfully removed and the bare surface looks perfect and smooth with terrain details remaining. Other surface fitting and smoothing techniques, such as the variable-order surface fitting (Besl and Jain, 1988), may be needed to provide an accurate smoothing depending on the terrain complexity. In addition, the raw lidar points within certain buffer of the raster DTM could be labeled as terrain points, which are used to create TINs thereafter.

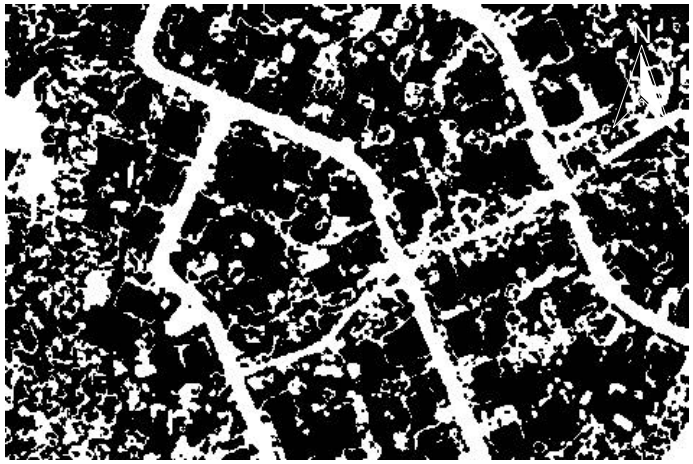
3.5 DTM GENERATION FOR BI-RETURN LIDAR DATA

The above methodology is proposed to generate DTMs for single-return range data. It has to be modified to utilize the information from first and last returns range and intensity data. The classification of intensity data can identify pavements on the topographic surface according to the separability of various materials on ground and the surface geometry; while bi-return range data ease the identification of non-terrain vegetated areas. This enables the simultaneous usage of geometric and radiometric information for analysis and processing of lidar data (Hu and Tao, 2003).

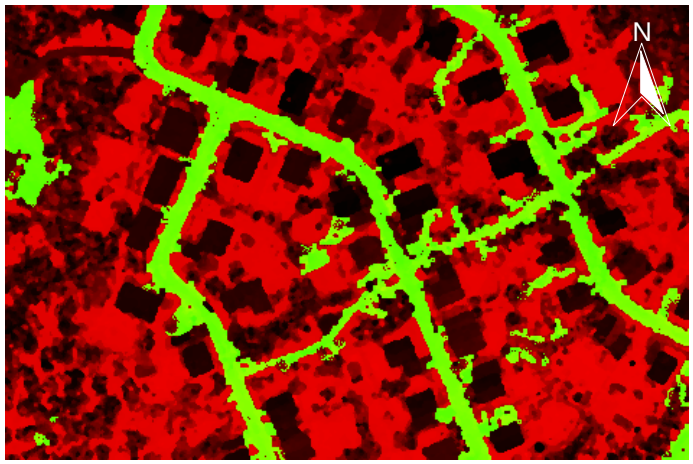
3.5.1 Identification of Road Points

Roads are usually coated by homogeneous materials (e.g., asphalt with pebbles or

concrete) over the natural terrain, have low heights, are in general very smooth, and are separated by distinct patches in either reflectivity (e.g., vegetated areas) or height relative to the bare surface (e.g., buildings). Road points can be identified based on four evidences in reflectivity, height, surface geometry and solidness.



(a) Road points classified using intensity image



(b) Road points overlaid on intensity image

Figure 3.7. Identification of road points (Village site)

- In *reflectivity*, the spectral signature of asphalt roads significantly differs from vegetation and most construction materials. The intensity image can be segmented using dual thresholds, denoted by $[Ti_l, Ti_h]$, which enclose the intensity range of the road class (see Section 4.3.1 for details).
- In *height*, roads are thin coats over topographic surfaces and thus have zero height in the DNM. But the height data itself is the byproduct of the expected DTM.

- In *surface geometry*, roads usually are composed of flat or sloped planes, and are very smooth surface patches.
- In *surface solidness*, roads are un-penetrable to laser beams, and thus have no difference between the first-return and the last-return range values. This evidence can be applied to multi-return lidar data to clean penetrable vegetation.

To this end, I apply to the intensity image and the current level of the pyramid an identification algorithm to detect road points by integrating reflectance property, surface geometry and smooth condition. The above evidences are expressed as four criteria used to identify road points.

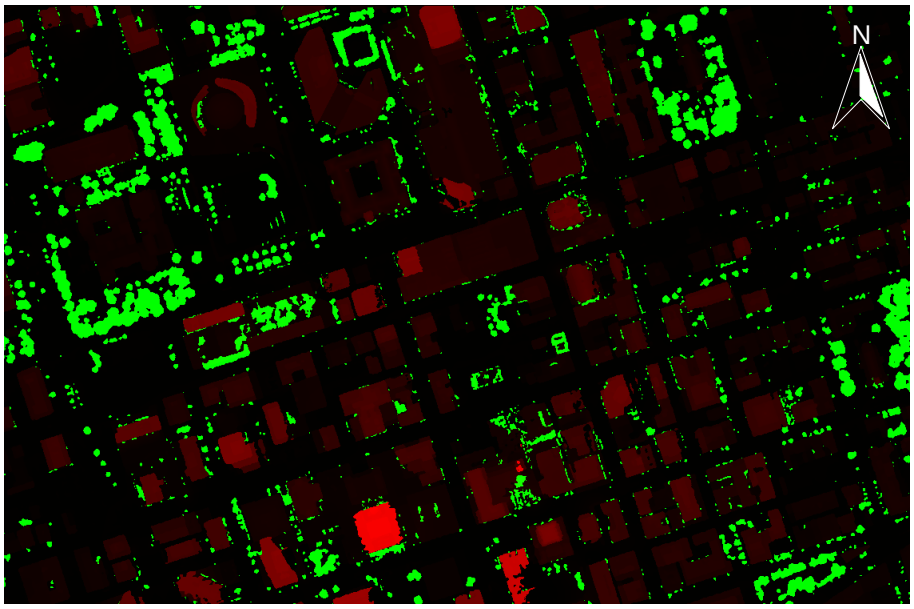
- *Reflectivity* condition. The point intensity is larger than Ti_l and less than Ti_h.
- *Height* condition. The smooth condition is used here instead of the stronger zero height constraint. Thus the point satisfies the smooth condition (i.e., Equation 3.3).
- *Geometry* condition. The point is on a flat or sloped plane (i.e., Equation 3.7c).
- *Solidness* condition. The elevation difference between the first and last returns elevation values is close to zero (see Section 3.5.2).

I manually select an intensity range (i.e., 31 to 42) to classify the intensity image for Village site (see Figure 3.7a); then perform a morphological opening to separate objects weakly connected to road ribbons and remove connected components smaller than 500 pixels; finally, the geometric and height conditions are applied to discard non-terrain points. Figure 3.7b shows the road points (green, in total 22149) overlaid on the intensity image (red). Similarly, other data may be used to classify terrain points on mountains. However, this helps little in forested or mountainous areas where no or only few roads are paved with asphalt or same materials.

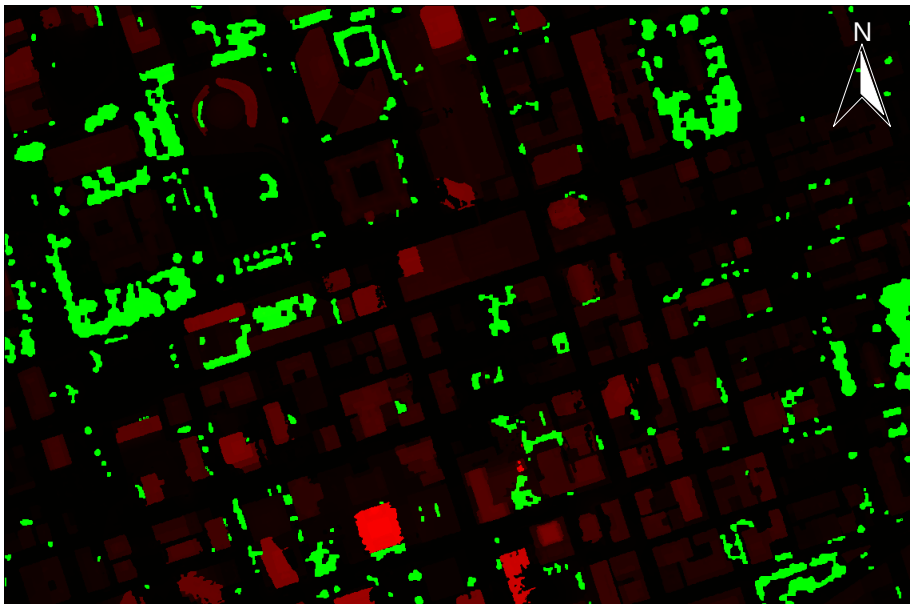
3.5.2 Derivation of Vegetation Height Models

The vegetation height model (VHM) is defined as the representation of penetrable vegetation (such as trees and bushes) and related properties such as height values (Hu and Tao, 2003). Separate returns of a laser pulse can be digitized when the vertical profile of

an object has multiple parts (e.g., the highest and lowest parts) within the laser footprint in decimeter level (see Section A.1.1). Recording of multiple returns is useful especially in case of trees. In the off-leaf condition, the last return for each pulse may hit on the ground. The ability of the laser to penetrate foliages makes it possible to distinguish vegetated areas from non-penetrable solid surfaces such as pavements, buildings and even grasslands. The VHM is generated using four steps.



(a) Rough VSM (green) overlaid on range image (red)



(b) Refined VSM (green) overlaid on range image (red)

Figure 3.8. Vegetation support models (Downtown Toronto)

- *Rough height model*. First, the *Z* value of each first-return point is replaced by the difference between its first-return and the last-return elevations, and the horizontal position of each point is taken from the first return. Then, the modified point cloud is transformed into an image. It should be noted that the grasslands have zero height.
- *Rough support model*. The rough height model is segmented using dual height thresholds, which are determined according to a priori knowledge about the height ranges of the lowest bush and the tallest tree, respectively. An iterative selection method (see Section B.1.6) is used to adjust these thresholds. Only those pixels whose heights are between these two thresholds are considered to be candidate high vegetation. This produces a binary representation of vegetation points, called vegetation support model (VSM), in which 1-valued pixels indicate the presence of penetrable vegetation. In Figure 3.8a, a rough support model (green layer) is generated for downtown Toronto.
- *Refined support model*. Because the footprint of a laser shot is in decimeter order when reaching the ground, multiple returns may happen at building walls and boundaries where elevations change very fast. In general, these flaw effects resulted from steep walls and complex roof structures of buildings appear as slim strips of 1 to 3 pixels wide (see green layer in Figure 3.8a). Morphological opening and closing operations (see Section B.1.3) are applied to eliminate these thin strips and tiny blocks that are possible speckles. The refined model is shown in Figure 3.8b (green layer). The DNM, if available, can be used to promptly discard points higher than the known tallest tree in the ROI.
- *Refined height model*. Since last returns do not always hit on the ground surface in forested areas, and thus the tree heights are likely underestimated if only the first and last returns data are used. The height values of vegetation points in the rough height model can be refined by retrieving values from the subtraction between the first-return DSM and the derived DTM for those 1-valued pixels in the refined support model (see Figure C.4 for an instance of Whitecourt area).

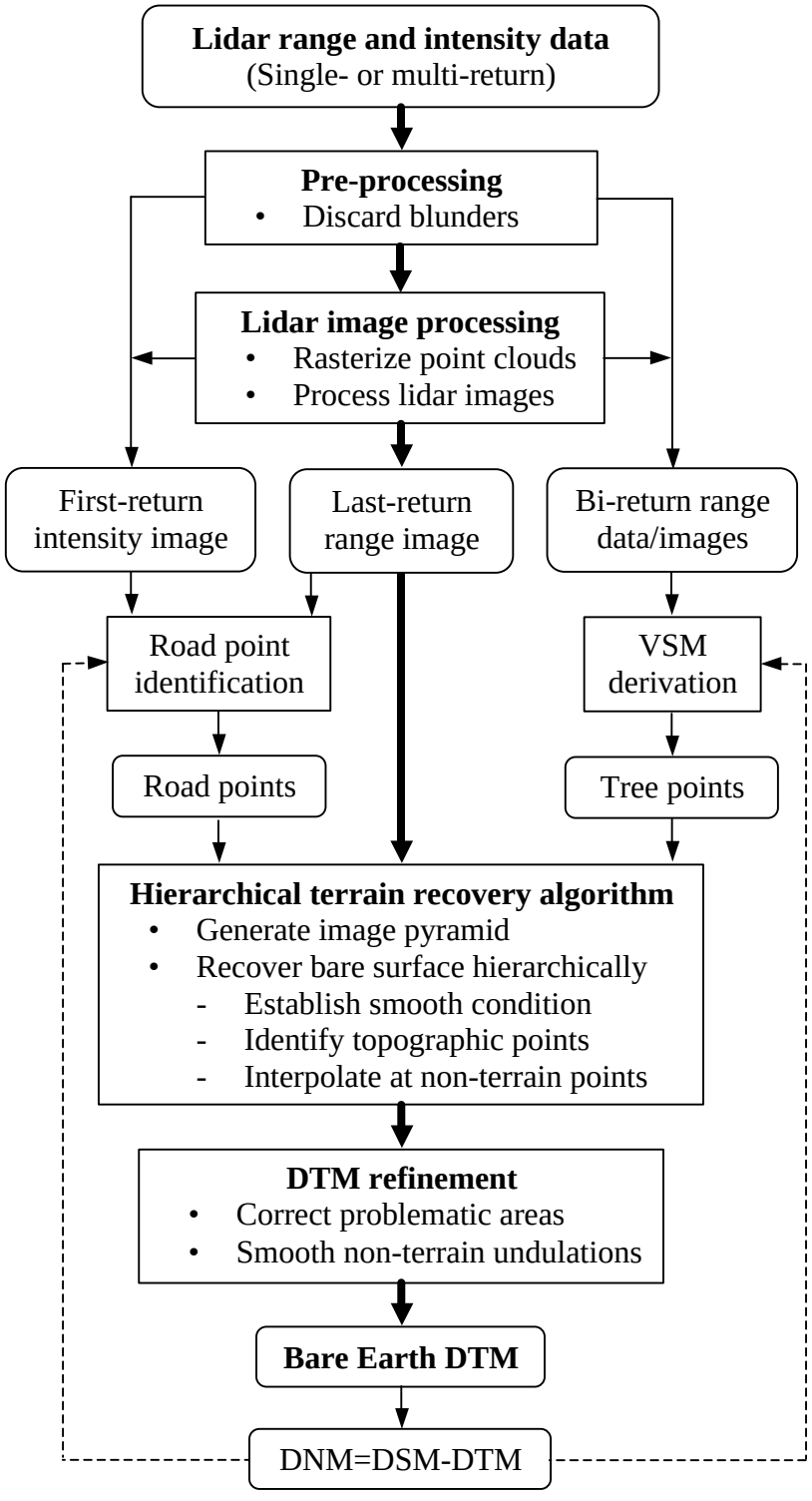


Figure 3.9. Flowchart of the hierarchical terrain recovery algorithm

3.5.3 The Modified Methodology

The workflow of the modified methodology for multi-return lidar data is illustrated in Figure 3.9, where the bold arrows give the workflow of the methodology using single-return data (see Section 3.2).

The HTRA is applied to generate DTM first using the (last-return) range image. It generates an image pyramid with varying detail levels from the raw range image by selecting local lowest points in image grids, and then recovers the bare surface hierarchically from the top level to the bottom level.

At each level, the terrain relief and slope information is used to establish the smooth condition, which determines the height thresholds used for screening out non-terrain points. In general, the relief threshold is usually selected when processing flat and mountainous areas; while the slope threshold is likely selected when processing building roof points and tree points. Part of non-terrain points can be identified rapidly when the VSM is available. The algorithm interpolates at these non-terrain points by fitting a first- or second-order polynomial using surrounding terrain points, including road points and other topographic points. More terrain points have to be identified using surface geometry if the road points are not enough at, for example, mountainous areas.

The processing is repeated until the bottom level is reached, and the bottom-level image is an initial estimation to the expected DTM. The VSM provides reliable evidence for reporting the presence of non-terrain points when processing the bottom level. For example, the parameter c_2 can be tuned automatically in windows containing tree and road points, so that the tree points must be classified as non-terrain ones and the road points must be classified as terrain points under the smooth condition.

Finally, the initial DTM is refined by correcting problematic areas and by moderate smoothing. An estimate of the DNM is the subtraction between the raw range image and the final DTM, and will be used for extraction of roads and buildings in next chapters.

Table 3.1. Information about five lidar datasets

Properties		**Village**	**Osaka**	**Toronto**	**Whitecourt**	**Santa Barbara**
Area type		Residential	Suburban	Urban	Forested	Mountainous
Laser returns		First	Last	First & last	First & last	First
Point number n_{ROI} (x10^6)		0.15	0.63	10.6	20.9	11.3
Coverage area A_{ROI} (km^2)		0.054	0.5	3.8	30	26
DSM Elevation range (m)	Raw	[158.68, 206.72]	[-20.34, 90.97]	[-399.78, 398.41]	[733.11, 970.19]	[0.31, 148.51]
	Filtered	[167.25, 205.61]	[30.65, 86.85]	[21.02, 343.51]	[733.54, 967.8]	[0.42, 147.72]
Intensity range	Raw	[1, 116]	[0, 6420]	[0, 8160]	[1, 640]	[0, 297.6]
	Filtered	[5, 86]	[0, 384]	[2, 199]	[2, 146]	[0, 163.2]

Table 3.2. Information related with DTM generation

Quantities	**Village**	**Osaka**	**Toronto**	**Whitecourt**	**Santa Barbara**
GSD s_c (m)	0.6	1	1	1.2	1.6
Image size (rows x columns)	315 x 474	744 x 656	2811 x 2292	3918 x 5318	3674 x 6971
Window size s_w (pixels)	28	99	120	100	75
Pyramid scale k	6	5	5	5	5
Pyramid levels n	3	4	4	4	4
DTM elevation range (m)	[167.3, 181.45]	[30.65, 46.36]	[21.07, 71.78]	[733.54, 943.34]	[0.42, 145.56]
Maximal DNM height (m)	26.7	47.3	301.52	26.73	58.39

3.6 EXPERIMENTS AND EVALUATION

To provide a comprehensive evaluation of the performance of the HTRA, I test five lidar datasets acquired at residential, suburban, urban, forested or mountainous areas, respectively. Table 3.1 gives the major information about these datasets.

3.6.1 DTM Generation Results

In Table 3.2, information related with the DTM generation procedure and the results is given for these five lidar datasets. In Figures 3.10 to 3.14, the contour lines are generated from the derived DTMs, and the profiles are delineated along the positions as indicated by dotted lines on original datasets showing the elevations in both the lidar DSMs (red) and the derived DTMs (green).

3.6.1.1 Village Site: Residential Area

Dataset. This is a sample data file provided with TerraSolid's TerraScan software (TerraSolid, 2001). In the village, the houses are larger but lower than the trees, and the buildings and trees are loosely sited. So this scene has low complexity. The data file is in simple ASCII format with three ground coordinates and intensity values for a single return. In Figure 3.10a, the raw lidar points have been classified into three classes, including bare surface (brown), building (red) and tree (green).

Results. Figure 3.10b shows the DTM with a three-time exaggeration in vertical direction. The light gray indicates the low elevation. The result looks good since the terrain elevations change gently in the whole study area. The maximum height of 26.7 m in the DNM occurs at a cluster of dense trees within the circle in Figure 3.10a, and this height may be the height of the tallest treetop.

3.6.1.2 Osaka City: Suburban and Urban Area

Dataset. The Osaka data was acquired in Osaka, Japan by Optech in 2002. There are bridges and viaducts. This scene is relatively flat. In the urban region, there are many small to large buildings, but few trees. The dataset is composed of 4 strips, and the

overlap between two adjacent strips is about 40%. Some points with extremely large intensity values are discarded during pre-processing. The lidar points are rendered by elevation as shown in Figure 3.11a, where the white strips in the scene are water areas.

Results. Figure 3.11b shows the generated DTM with exaggeration of four times in vertical direction. The quality of the DTM is good. The bridges and viaducts are removed, and most water bodies are kept. The small terrain relief of the DTM indicates that the test field is very flat, but the terrain coverage appears more complex. The maximum height in the DNM is 47.3 m, and it occurs at one tall building within the ellipse in Figure 3.11a.

3.6.1.3 Downtown Toronto: Urban Area

Dataset. The bi-return lidar dataset was collected at downtown Toronto using ALTM 3020 in early 2002. This is a typical urban scene. This study area has relatively gentle variations in elevation. It contains many big buildings that are larger and taller than surrounding trees, viaducts and cars on roads and parking lots. The dataset consists of six strips all along the north-south direction. In total, about 66 thousands of blunder points are eliminated during pre-processing. The filtered lidar points rendered by elevation are shown in Figure 3.12a, where the building enclosed in the circle is First Canadian Place [6]. The Map Library [7] at York University provided four digital aerial color DOQs and a hardcopy of the topographic map (in the scale of 1:50,000) covering the study area. The index number of the map is 30M/11 in the National Topographic System.

Results. The largest window size is 306 m when using the area and perimeter properties because multiple large buildings are melt together. To reveal the terrain details, the DTM is visualized in 3-D with 5-meter contours in Figure 3.12b. The buildings, trees and cars are all removed correctly. The overall terrain surface is like a sloped plane, and the elevation is reduced gradually from 72 m at north to 21 m at south. The pits are found to be water pools or subway entrances by an investigation in field. The maximum height of 301.52 m is located on the roof of the First Canadian Place.

---

[6] First Canadian Place is 298 m high at its roof and reaches 355 m with a structure on its roof. It has been the tallest building in Canada since 1975 (see URL: http://www.skyscrapers.com/re/en/wm/bu/112676/).
[7] Ms. Trudy Bodak (Email: tbodak@yorku.ca) is the Librarian from Map Library, York University.

3.6.1.4 Whitecourt: Forested and Mountainous Area

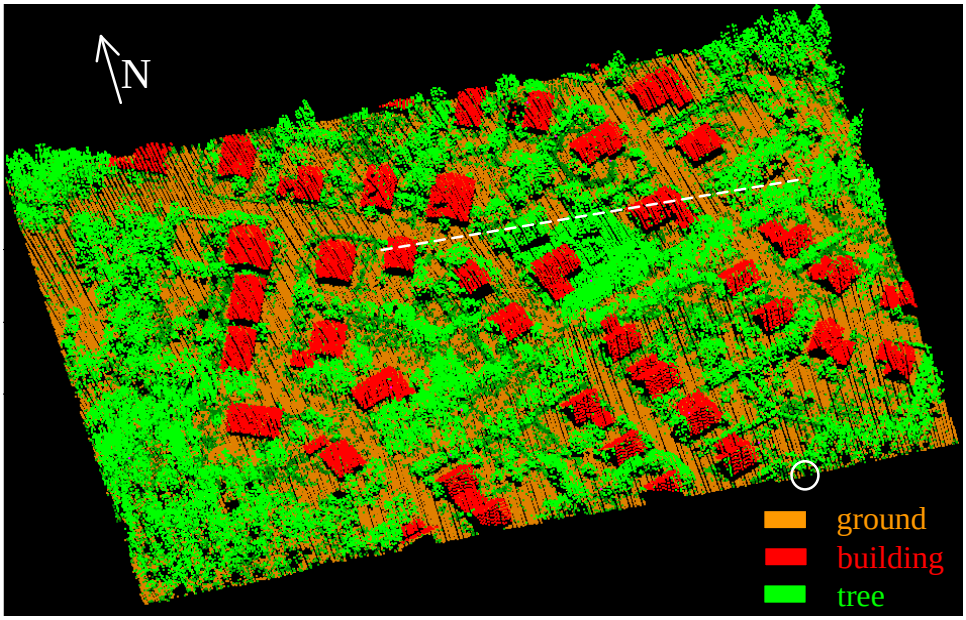
Dataset. The bi-return lidar data was collected in August 2002 along a pipeline near Whitecourt, Alberta, using an ALTM 2050. This is a typical forested area with large terrain relief. Dense trees cover most area since the data was acquired in summer. The pipeline runs SE to NW through the centre of the region with large elevation changes. A ridge crosses the northwest portion. A patchwork of clear-cut was done to remove trees within about 42 meters along both sides of the pipeline's route. In Figure 3.13a, the lidar points are rendered by elevation. The Map Library [7] provided the hardcopies of the two adjoining topographic maps (in the scale of 1:50,000) covering the study area. Their index numbers are 83J4 and 83J3 in the National Topographic System.

Results. Figure 3.13b shows the generated DTM that is exaggerated two times in elevation. The lighter the gray, the lower is the elevation. The middle portion is highest, and this results in elevation difference along the pipeline route. To successfully remove tree points close to the terrain surface, the parameter c_2 is tuned to be 0.6.

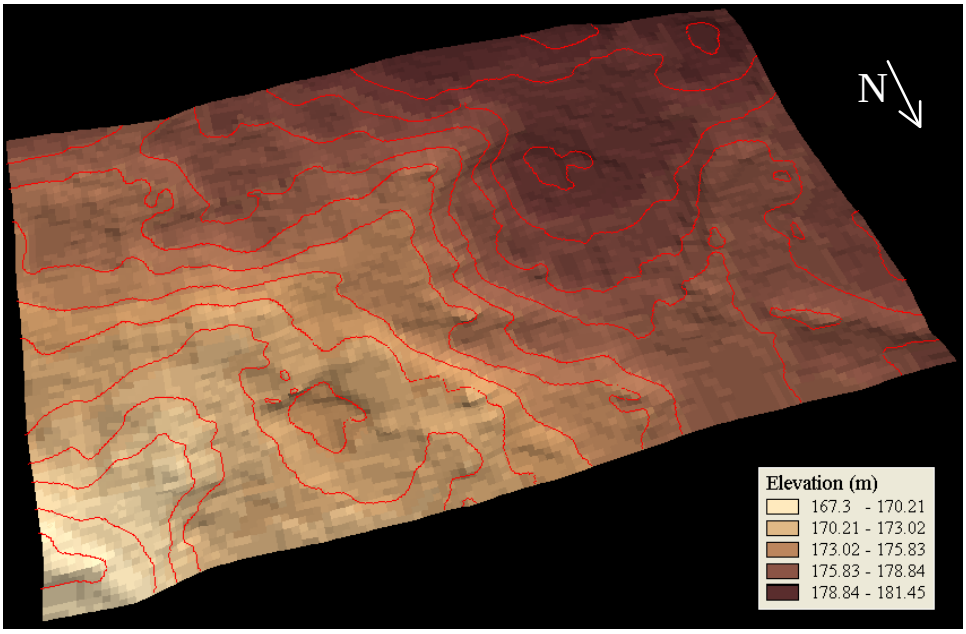
3.6.1.5 Santa Barbara Airport: Mountainous, Suburban and Forested Area

Dataset. This area has buildings, houses and dense forests. In between trees, the houses are smaller and lower. In the center, the airport has three runways and very large waiting rooms. At the north part, there are hills with large undulations. The lidar dataset has two parts, between which there is a void narrow strip about five meters wide. The point density at the upper portion is larger than that at the middle portion, and the bottom portion is water body. Figure 3.14a shows the lidar points rendered by elevation. An aerial color DOQ with a GSD of 1 m is also available, and covers the whole region.

Results. The generated DTM is shown in Figure 3.14b, where the light gray indicates the low elevation. The void stripe is filled. Buildings and forests are removed correctly, and the mountains at the north portion are successfully retained. At the highest mountain peak as enclosed in the circle in Figure 3.14a, the peak elevation in the DTM is 145.56 m, and is lower by 2.16 m compared to the true peak elevation of 147.72 m in the DSM.



(a) Raw lidar data



(b) The DTM with 1-m contours

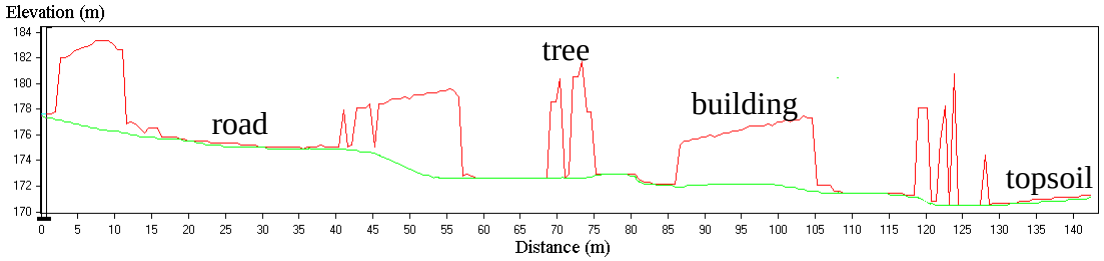
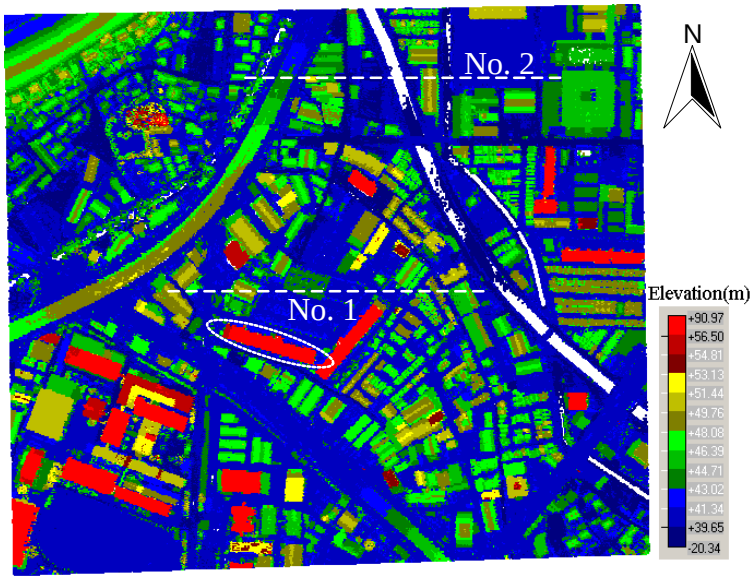
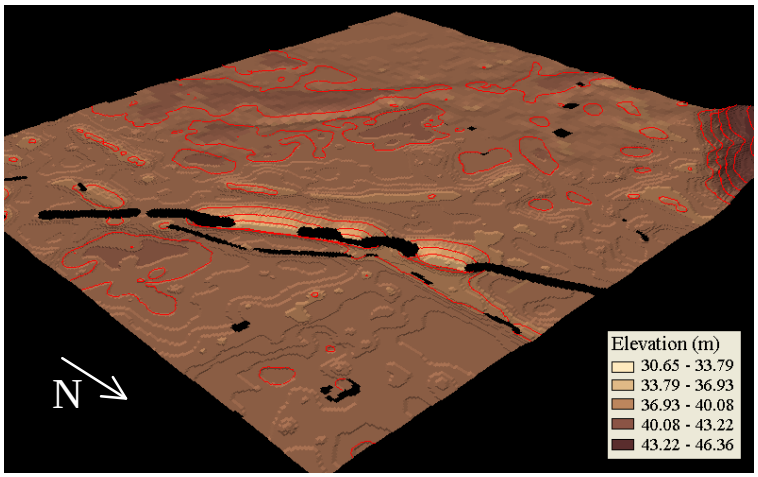


Figure 3.10. Village site (residential area)



(a) Raw lidar data



(b) The DTM with 2-m contours

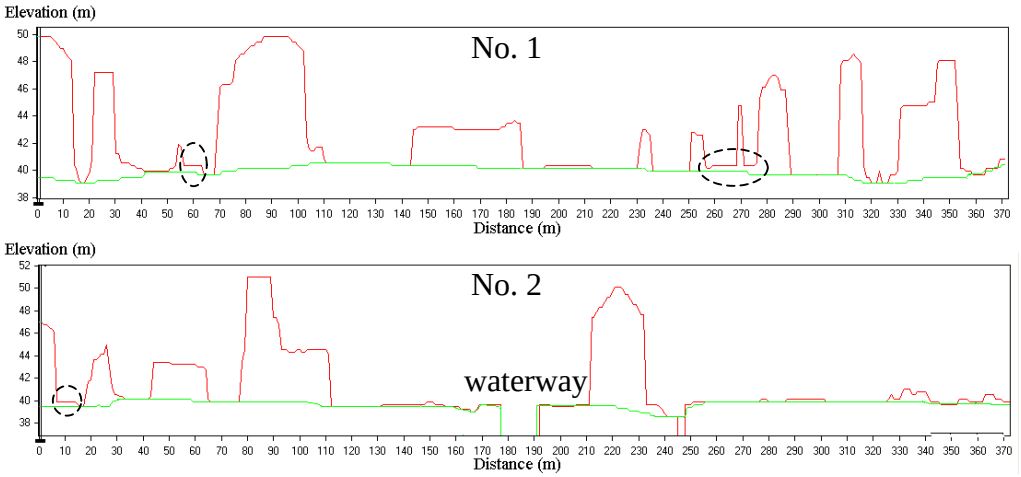
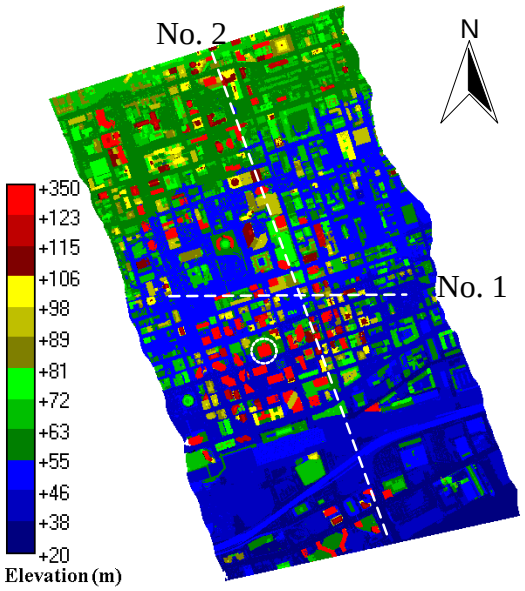
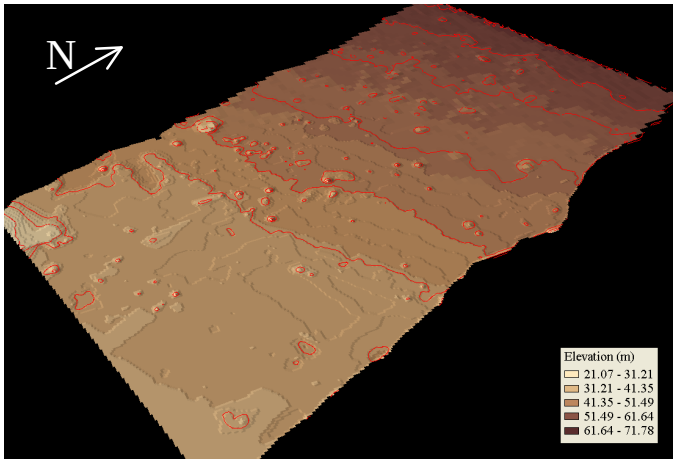


Figure 3.11. Osaka city (suburban and urban area)



(a) Filtered lidar data



(b) The DTM with 5-m contours

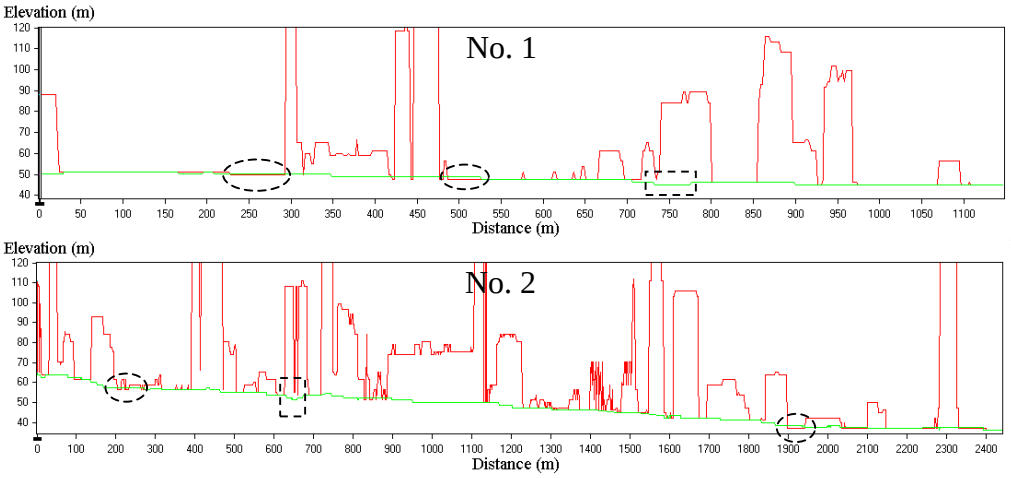
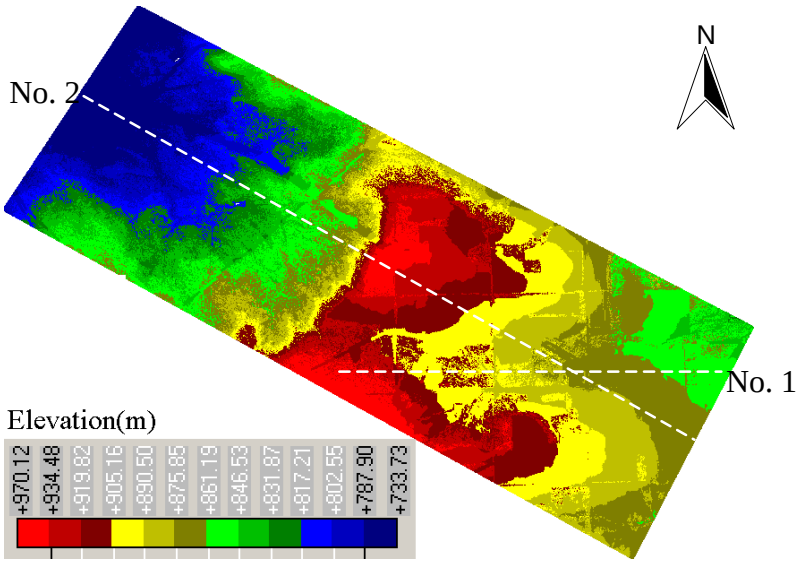
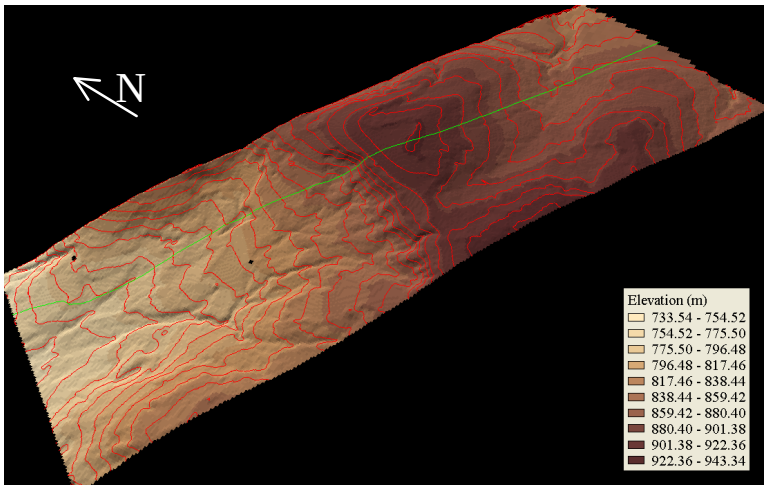


Figure 3.12. Downtown Toronto (urban area)



(a) Raw lidar data



(b) The DTM with 10-m contours

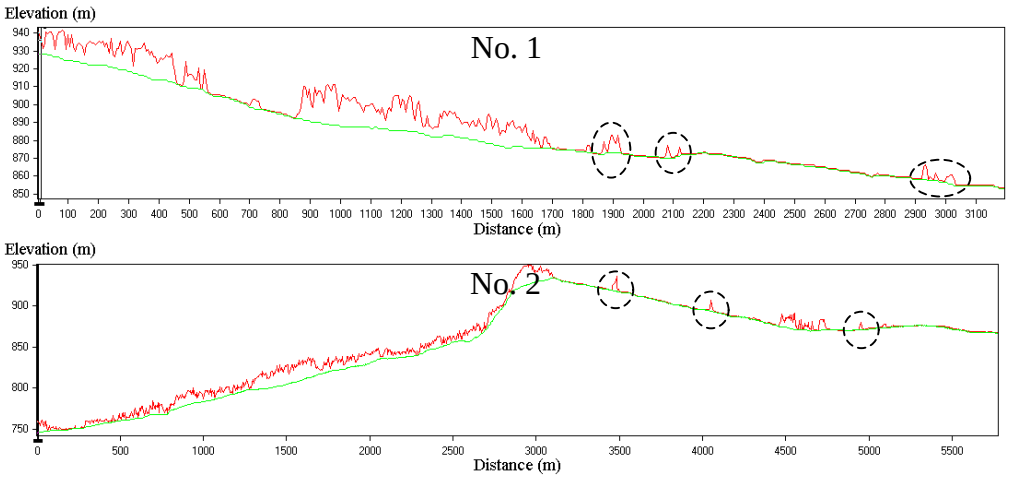
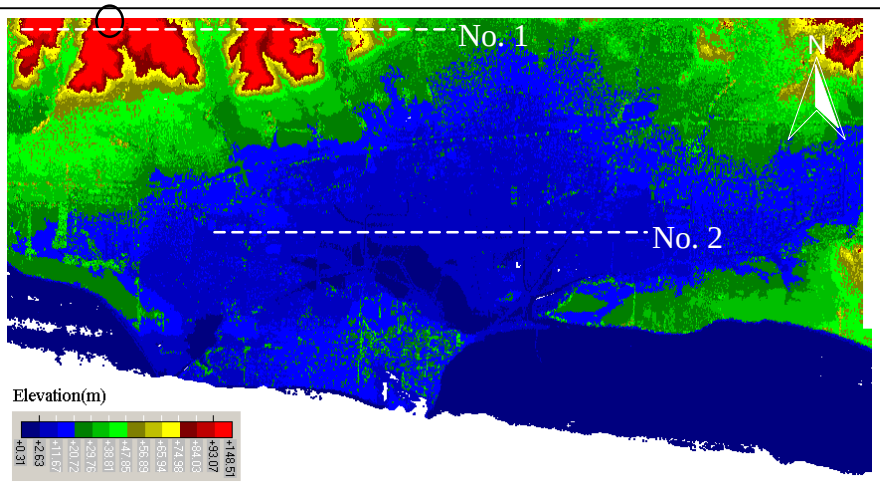
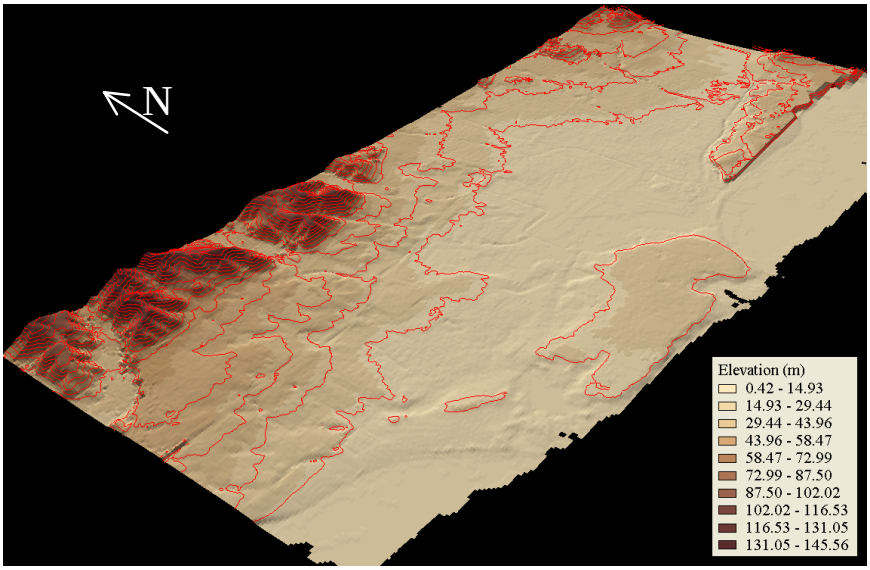


Figure 3.13. Whitecourt pipeline (forested and mountainous area)



(a) Raw lidar data



(b) The DTM with 10-m contours

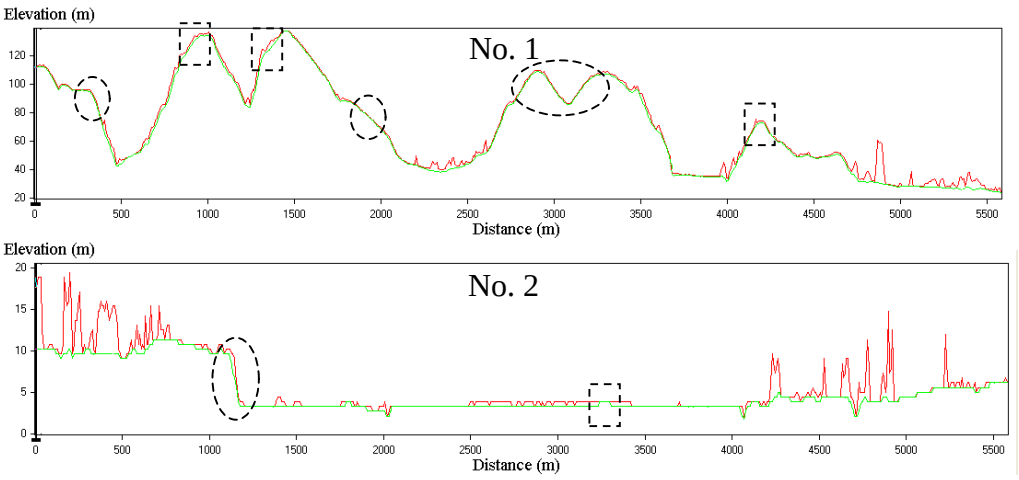


Figure 3.14. Santa Barbara airport (mountainous, suburban and forested area)

3.6.2 Quality Assessment

The quality of the experimental results and the performance of the HTRA are assessed from following four aspects:

Profile analysis

It is observed that the derived DTMs are close to the expected bare Earth surfaces as can be visually inferred from corresponding DSMs in Figures 3.10 to 3.14. I will make a closer exam by analyzing a series of representative profiles.

- *Village site* (Figure 3.10). The profile has gentle terrain relief, and crosses buildings, topsoil, roads, trees and bushes.
  - The DTM runs smoothly on roads and topsoil, and under buildings and trees. Buildings and trees in the scene are all removed correctly.
  - The DTM patches interpolated under buildings and trees bridge their surrounding roads and topsoil smoothly since the bare surface has little elevation variations.
  - The DTM is slightly under-estimated at some building and bush surroundings, is perfectly estimated at tree surroundings, on roads and on topsoil.

- *Osaka city* (Figure 3.11). The first profile crosses buildings, roads and trees in the scene center, and the second profile crosses buildings, trees, viaducts and waterways.
  - In both profiles, the terrain relief is small. Buildings and trees are all removed correctly, and the interpolated patches under buildings and trees bridge their surroundings well.
  - In both profiles, the DTM is under-estimated up to about 0.5 m as enclosed in the ellipsoids. This is because some road points are not identified correctly when classifying intensity data and are discarded as non-terrain points. Then their elevations are interpolated from lower surroundings.
  - In the second profile, the wide waterway is kept correctly, but the narrow one is filled up. But if the void areas are not filled to some gap, the occluded building surroundings cannot be recovered.

- *Downtown Toronto* (Figure 3.12). The first profile crosses low to tall buildings, roads, and trees horizontally, and the second profile crosses low to tall buildings, trees, and viaducts vertically in the scene center.
  - In both profiles, buildings and trees are all removed correctly, and the interpolated patches under buildings and trees bridge their surrounding well in most segments.
  - In both profiles, the DTM is over-estimated up to about 0.5 m as enclosed in the ellipsoids. The reason is that the outdoor parking lots beside are full of cars.
  - In both profiles, the DTM is under-estimated up to about 2 m under the building as enclosed in the rectangles. The reason is that some low points in those areas are not removed in the pre-processing step. In the second profile, this results in lower elevations partly due to the use of second-order polynomials.

- *Whitecourt* (Figure 3.13). Both profiles have large elevation changes, crossing topsoil, dense trees and grasslands. This scene contains flat to mountainous areas. The last return may not hit on the ground under trees since the dataset was collected in summer when the foliage is most dense in a year.
  - In both profiles, the DTM runs on the topsoil perfectly, and the thin tree stripes enclosed in the ellipsoids are removed correctly.
  - In the first profile, the DTM runs smoothly from the summit down the flat region.
  - In the second profile, the elevation difference in the middle is about 27 m. I am not sure if the summit has been cut wrongly since trees may be taller than 27 m.

- *Santa Barbara airport* (Figure 3.14). The first profile has large elevation changes crossing bare mountains, valleys, trees and buildings; the second profile crosses large buildings, trees, runways, waterways and waiting halls at the airport.
  - In the first profile, the fidelity of mountain ridges, valleys and slopes enclosed in the ellipsoids are retained well. This is due to the use of second-order polynomials when performing the interpolation.
  - In the first profile, the DTM runs closely under mountains. The summits and ridges enclosed in the rectangles are cut a lot up to about 2.5 m. This is because

the largest window size (i.e., 120 m) is too large, resulting in the difficulty to identify topographic points based on those local lowest points at discontinuities where elevations change abruptly.

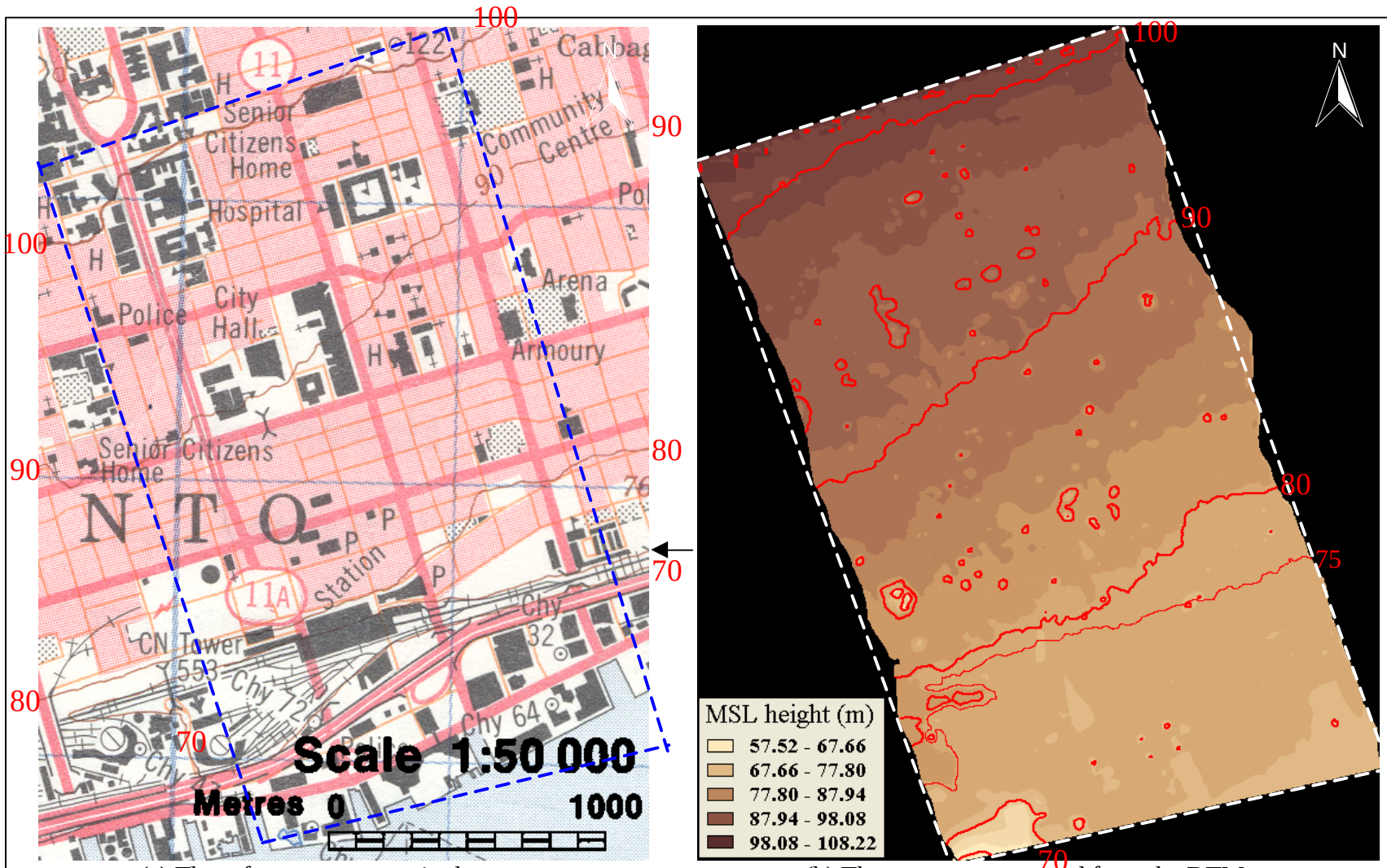
- In the second profile, the steep ramp enclosed in the circle is smoothed to some extent. The top edge is about 4 m lower than its observed elevation. To conquer this drawback, extra data such breaklines have to be used to limit the interpolation to either side.
- In the second profile, the DTM is over-estimated at the airport waiting halls as enclosed in the rectangle. The size in any dimension of the waiting halls is larger than 120 m. The result is that the roof points in its middle are treated as terrain points at the top level of the pyramid, and are kept during the whole procedure.
    * One way to solve this problem is to increase the largest window size to 200 m. but this will result in another drawback occurred in the first profile. That is, the mountains will be under-estimated more severely.
    * Another way is to divide this scene into multiple overlapping sections so that each section is a scene with lower complexity with homogenous terrain and coverage types. Then different largest window sizes are used for different scenes. But this method has not been evaluated.

Contour analysis

As shown in Figures 3.10 to 3.14, the derived contours are smoother in areas with fewer buildings and other constructions, and have more undesirable isolations and depressions especially at built-up areas in cities.

Table 3.3. Geoid heights relative to WGS84 ellipsoid (downtown Toronto)

Point number	**1**	**2**	**3**	**4**	**5**
Latitude (North)	43^{o} 39.8’	43^{o} 39.8’	43^{o} 38.3’	43^{o} 38.3’	43^{o} 39.05’
Longitude (West)	79^{o} 23.6’	79^{o} 23.2’	79^{o} 23.2’	79^{o} 23.6’	79^{o} 23.4’
Geoid height (m)	-36.438	-36.439	-36.458	-36.457	-36.449



(a) The reference contours in the map

(b) The contours generated from the DTM

Figure 3.15. The reference and generated contour lines (downtown Toronto)

- *Downtown Toronto*. Figure 3.15a shows the topographic map produced from aerial photographs taken in 1980 and the dashed quadrangle enclosing the ROI. In the map, the elevations are in meters above mean sea level (MSL) referenced to North American Datum of 1927 (NAD27), and the contour lines have the interval of 10 m. But the elevations in the lidar dataset are referenced to WGS84 ellipsoid. First, I measured the geographic coordinates of the four corners and the center point of the ROI on the map, and used GPS·H [8] to calculate their geoid heights referenced to WGS84 ellipsoid (see Table 3.3). The accuracy of the calculated geoid heights is 5-cm LE95 in the southern regions of Canada. The differences among these geoid heights are at centimeter level, and are significantly smaller than the contour interval. So I simply shifted all the grid points in the derived DTM vertically by subtracting the average geoid height (i.e., -36.4482 m) to convert the ellipsoidal elevations to heights above MSL. Then, contour lines in the interval of 10 m were generated for the shifted DTM (see Figure 3.15b).
  - At the heights of 80 m, 90 m and 100 m, the generated and reference contours are situated at close locations and represent similar trend of terrain relief overall with some differences.
  - There are unreasonable oscillations along the generated contours. They are situated in the six overlapping strips oriented in north-south direction, and show discrepancies in elevation. To reduce this effect, a supplemental block adjustment of lidar strips is needed. Also, the automated generation of contours cannot work better than the manual compilation in photogrammetry.
  - The reference contour at the height of 70 m as indicated by the arrow in Figure 3.15a looks like a tripe and does not present in the generated ones. The contour at the height of 75 m was generated and shown in Figure 3.15b (also see Figure 3.12b). It is observed that this auxiliary contour loses its way at the bottom. Two reasons result in this effect:

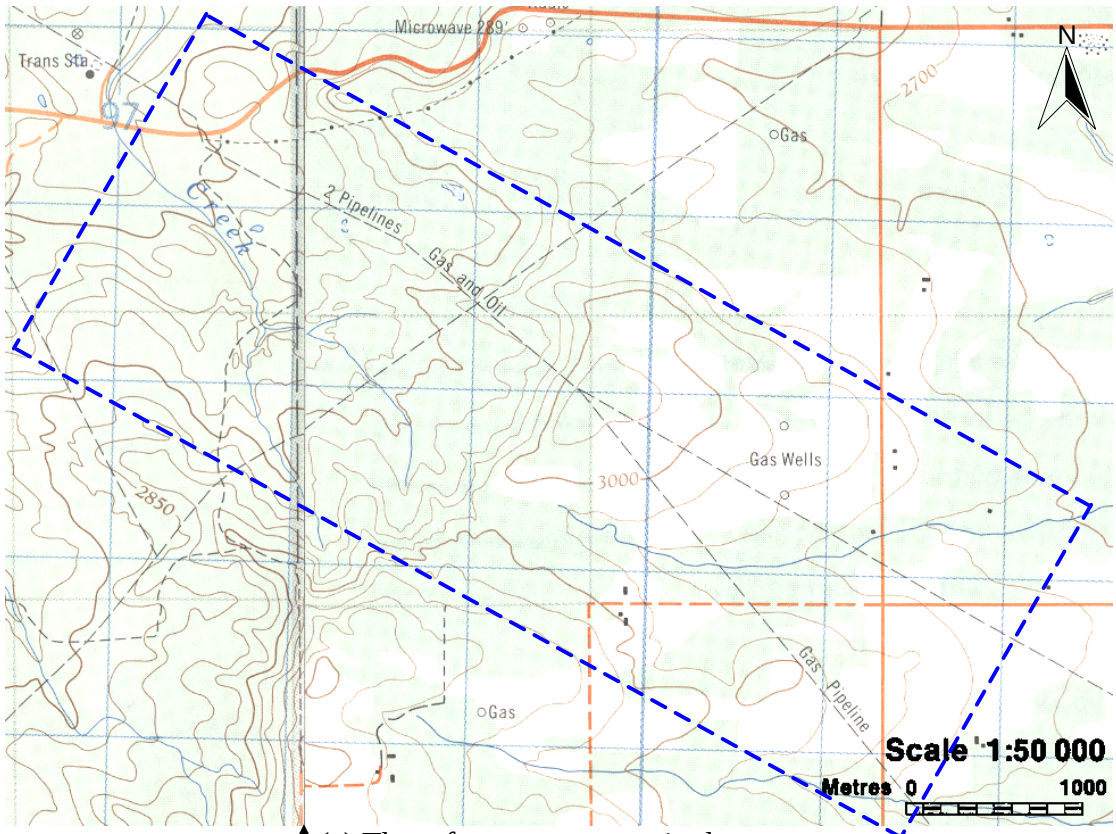
[8] GPS·H is a height transformation package supplied by the Canadian Geodetic Survey Division. It allows GPS and DGPS users in Canada to convert NAD83 (CSRS98) or ITRF96 ellipsoidal heights to heights above mean sea level. WGS84 was compatible with NAD83 in the past, but has been redefined and it is now compatible with ITRF (URL: http://www.geod.nrcan.gc.ca/).

  * Lidar points reflected from the neighboring viaducts/overpasses are removed as non-terrain points and the terrain surface under those objects were interpolated from points reflected from the surrounding objects.
  * The terrain along that strip is over-interpolated because the metro station therein is partly attached to ground as also can be observed from the right circle in the second profile in Figure 3.12.
- The DTM has a few isolations and more artificial depressions. Most depression contours are at water pools (large depressions) and subway entrances (small depressions). This is because points lower than the feature surroundings are reflected and are not removed in the pre-processing step.
- The generated contours at the bottom are due to water features in Ontario Lake. Many lidar points may be reflected from the boats parking in that region (by a field investigation), and thus present higher elevations. But these points cannot be reliably removed if the water boundaries are not used.

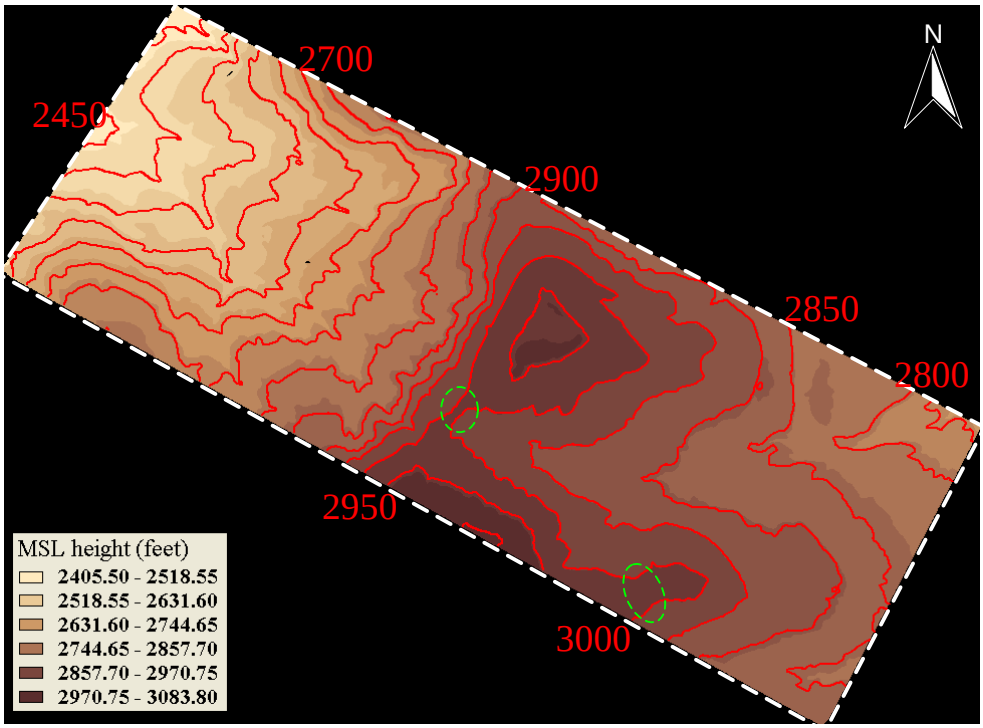
Table 3.4. Geoid heights relative to WGS84 ellipsoid (Whitecourt)

Point number	**1**	**2**	**3**	**4**	**5**
Latitude (North)	54^{o} 6.5’	54^{o} 7.5’	54^{o} 5.9’	54^{o} 4.9’	54^{o} 6.2’
Longitude (West)	115^{o} 31.6’	115^{o} 30.6’	115^{o} 25.9’	115^{o} 24.9’	115^{o} 28.25’
Geoid height (m)	-17.911	-17.942	-17.976	-17.978	-17.938

- *Whitecourt*. In Figure 3.16a, the topographic map is mosaicked from the two adjoining maps (left: 83J4; right: 83J3), both of which were produced from aerial photographs taken in 1978, and the arrow indicates the boundary; the dashed quadrangle encloses the ROI. In the map, the elevations are in feet above MSL referenced to NAD27, and the interval of the contour lines is 50 feet. Similarly, I measured five points in the map (see Table 3.4), and shift the derived DTM vertically by subtracting the average geoid height (i.e., -17.949 m) to convert the ellipsoidal elevations to heights above MSL. Multiplied by 3.28, the heights in meters were converted to values in feet. Then, contour lines in the interval of 50 feet were generated for the shifted and scaled DTM (see Figure 3.16b).



(a) The reference contours in the map



(b) The contours generated from the DTM

Figure 3.16. The reference and generated contour lines (Whitecourt)

- The generated contours have similar shape but are smoother compared to the reference ones. Most of them are situated at positions close to their counterparts in the map. But the DTM is not good in geomorphological detail upon a close inspection. For example, the three contours at the height of 3000 feet are separate in the map, but are wrongly connected at places enclosed in the ellipsoids as shown in Figure 3.16b. There are three reasons:
  * The last return lidar points may not hit on the ground at densely wooded areas, thus losing information about some geomorphological structures. The result is that generated contours at those areas will likely lose their way.
  * Some important geomorphological structures (e.g., ditches and ridges) have been smoothed or even eliminated by the algorithm.
  * The reference contours produced from aerial photographs may be not very accurate or reliable especially at densely wooded areas since only a small number of ground points are visible on stereo pairs. Moreover, the ground may be covered with very dark shadows that prevent accurate measurements in the passive photogrammetry.
- Along water streams and valleys, the generated contours present more similar shapes as their counterparts in the map because there are fewer trees grown along them as observed in Figure 3.13.

Accuracy analysis

I check the relative elevation differences between the raw range images and the derived DTMs at building surroundings, tree surroundings, mountains, roads, grasslands and other open areas. The checkpoints are selected with even distributions in the ROIs. The quality of the derived DTMs under buildings and trees can be checked by inferring from their surroundings. This is because the surroundings are used to interpolate the terrain surfaces under building and trees in my algorithm. If buildings and trees are not removed correctly, their surroundings are likely over-interpolated. The quality at bare areas, including mountains, wide roads, and open areas, will check the capability of retaining terrain details at bare surfaces.

These areas are located visually from the range and intensity images, and, if available, DOQs. The accuracy check is equivalent to measure the height values in DNMs. Table 3.5 gives some statistics about the overall accuracy for above five datasets. The accuracies at building and tree surroundings are given in Table 3.6. The Whitecourt dataset is not included because it contains no buildings and it is difficult to visually discriminate if a checkpoint is beside a tree or not in dense forest. The accuracies at bare areas are better than –4.5-cm mean error and 10-cm Std in the first four datasets, and –8-cm mean error and 11-cm Std in Santa Barbara dataset.

Table 3.5. Overall accuracies for generated DTMs

	Village	**Osaka**	**Toronto**	**Whitecourt**	**Santa Barbara**
Check points	40	60	100	120	152
Mean (cm)	-2.2	-4.4	-1.8	-3.9	-7.9
Std (cm)	5.3	7.9	9.9	7.6	10.9
RMSE (cm)	5.8	9.1	10.1	8.6	13.4
LE90 (cm)	9.9	15.7	18	13.5	23.2
Worst error (cm)	-17	-56	+50	-39	-253

Table 3.6. Accuracies at building and tree surroundings

	Village	**Osaka**	**Toronto**	**Santa Barbara**
Check points	8	21	32	20
Mean (cm)	-7.6	-8.5	-5.7	-8.3
Std (cm)	10	11.5	14.6	12.4
Worst error (cm)	-17	-56	50	-69

- The DTMs have good accuracies at feature surrounds and bare areas. The worst-case error, at the confidence level of 90%, is below 25 cm in the above five datasets.
  - The DTMs are slightly under-estimated in overall, but are under-estimated more severely at building and tree surroundings in most situations (except downtown Toronto dataset). The reason resulting in the under-estimation is that the smooth

condition has been configured to minimize Type II error and the final refinement also has removed all the possible over-estimations at checked areas.

- The accuracies at bare areas are very close to the range resolution (i.e., 5 cm) in the first four datasets. It is believed that these errors are due to the smoothing of those detailed undulations on the bare terrain surfaces.
- The accuracies at feature surroundings are inferior to those at bare areas in the first three datasets. This is reasonable since there are often many small structures and objects attached or detached to those features at their surroundings.
- The accuracies are worst at mountain areas, and are –57.1-cm mean error and 29.6-cm Std in Santa Barbara dataset. This has led to worse accuracies at bare areas than other datasets. The reason is that the mountain summits, ridges and steep ramps may be under-estimated a lot.

• Some researchers reported the quality evaluation of derived DTMs. But they have used significantly smaller datasets.
  - Axelsson (2000) reported absolute and relative accuracies for two datasets. It is repeated here that his algorithm is implemented in TerraScan ™ - the most widely used COTS software for lidar data processing.
    * The absolute accuracies are 19-cm mean error and 8-cm Std for dense bushes in the Kymlinge dataset (with 216k points in total).
    * The relative accuracies are 2-cm mean error and 6-cm Std for flat areas, and are 14-cm mean error and 3-cm Std for mixed areas (with bushes, roads and paved areas) in Arlanda airport dataset (with 273k points).
    * The relative accuracies are 9-cm mean error for road surface, and 2.6-cm mean error for dense forested areas in the Kymlinge dataset.
  - My results are close to the relative accuracies reported by Axelsson (2000) in overall, and the performance changes with respect to different areas.
    * My algorithm often under-estimates the bare Earth since the mean errors are negative, while Axelsson's always over-estimates it resulting positive errors.
    * My algorithm obtains slightly less mean errors than Axelsson's at road surfaces (-2.2 cm vs. 9 cm) and mixed areas (-2.2 cm to –7.9 cm vs. 14 cm),

but obtains slightly worse accuracies at flat areas (-4.4 cm vs. 2 cm) and densely forested areas (-3.9 cm vs. 2.6 cm).

  * My algorithm has larger standard deviations than Axelsson's at flat areas (7.9 cm vs. 6 cm), mixed areas (5.3 cm to 10.9 cm vs. 3 cm). This shows that the algorithm needs an improvement to work more consistently.

- No experimental results of lidar-derived DTMs have been reported for lidar datasets acquired at mountainous areas. But in Wang et al. (2001), the DTM derived from the IfSAR data with a 5-m GSD, which is acquired at Morrison, Alberta is compared with the bare surface that is manually edited from a lidar dataset covering the same region. The vertical accuracy is about 2.2-m RMSE at mountainous areas. My results are better since the vertical accuracy is 0.64-m RMSE at mountainous areas in Santa Barbara dataset. But it is should be noted that the vertical accuracy is 1-m RMSE and 0.2-m RMSE, respectively in the IfSAR and lidar datasets.

Algorithm analysis

The performance of the HTRA is subject to its component techniques and their efficient combination, and is analyzed based on the above qualitative and quantitative assessments.

- My algorithm is suited to process complex scenes sampled in different densities under a uniform technical framework. No other algorithms have reported such a capability.
  - The testing datasets are comprehensive and are acquired at residential, urban, forested and mountainous areas, which contain different combinations of many typical terrain and coverage types.
  - The point density of these datasets changes from 0.43 to 2.8 points/m^2. This shows that the algorithm is suited to process median- to high-density data.
  - The algorithm works well at feature surroundings and bare grounds. But the derived DTMs are under-estimated. A part of errors are due to two factors: the interpolation and smoothing operations at places, such as ridges, where discontinuities on the bare Earth occur; and the smoothing of detailed terrain

undulations. To avoid the smoothing along geomorphologic features, breaklines and spot heights from photogrammetry have to be introduced.

- The last-return data is more preferable than the first return as the tests show that the former can be used to generate slightly better results at forested areas.
- The algorithm has to be improved to reliably avoid eliminating lidar points that are situated on steep embankments, slopes and cliffs to retain the geomorphological detail. One way to automate this processing is to incorporate ancillary data such as breaklines and/or spot heights into the DTM generation workflow. But this has not been implemented.
- If a dataset contains many blunders such as very low or high points and unreliable points reflected from water features, the algorithm cannot produce reliable results at neighboring regions. Therefore, the preprocessing needs more improvements, for example, using water boundaries to remove reflections from water features.

• The algorithm effectively combines multiple component techniques: the hierarchical approach, smooth condition, data fusion, and interpolation methods.

- The hierarchical approach enables the gradual addition of denser terrain points and the recursive refinement of knowledge about ground coverage since it is only possible to find a part of terrain points at a first glance. Therefore, the algorithm is robust and efficient, and can be adapted to complex contents in one scene.
- The smooth condition intelligently integrates both terrain relief and slope information in a single formula. Therefore, the algorithm can intelligently distinguish between terrain and non-terrain points in a single pass, and also minimizes the overall misclassification. It also adaptively changes with different sampling densities and different resolutions at different pyramid levels. The rational is that larger thresholds can minimize Type I error when less feature details are sampled and more terrain points with relatively lower elevations have been identified.
- A typical scene often contains trees and/or roads. The use of multi-return range data can reliably identify tree points, and intensity data can be used to identify road points with the aid of evidences in height, surface geometry and solidness.

This has three advantages: the reliable deletion of non-terrain points in the VSM, the addition of denser terrain points, and the adaptive adjustment of the threshold parameters in the smooth condition. Therefore, the algorithm can produce good results for scenes with trees and/or roads using multi-return data.

- First- or second-order bivariate polynomials can approximate natural terrain facets and have an additional function of moderate smoothing. The identification of denser topographic points makes the interpolated surfaces keeping the fidelity of natural terrain especially in mountainous areas. But first-order polynomials are more stable than second-order ones.

• The algorithm is based on a hypothesis of finding initial terrain points. That is, lowest points at the top level of the pyramid are assumed to be on the bare Earth.
  - The use of local lowest points as the first set of terrain points may result in Type I error at terrain discontinuities, which may make the bare Earth under-estimated. The later interpolation and smoothing operations remove some terrain details such as road curbs.
  - This hypothesis may fail if the above assumption is not fully satisfied in the following two conditions.
    * The largest window size (i.e., the grid size at the top level of the pyramid) is smaller than the smaller dimension of some object in the scene. In this situation, Type II error will be very severe near those objects since some object points are wrongly accepted as terrain points in the coarse DTM, and this error cannot be recognized in subsequent processing steps. This failure can be eliminated if the largest window size is increased appropriately to be able to see on the terrain at any grid of the top level.
    * At densely forested areas, lidar beams seldom penetrate the foliages and thus cannot hit on the terrain surface. In this situation, the algorithm has no way to accurately estimate or predicate the bare Earth surface beneath the tree canopies, and always produce an over-estimated DTM. It is recommended that the data acquisition be done in off-leaf condition for forested areas.

3.7 SUMMARY

I developed the HTRA to generate DTMs for single- or multi-return range and intensity data. The algorithm can intelligently discriminate between terrain and non-terrain points for complex scenes, and has been tested using multiple median- and high-resolution lidar datasets. The results show that the non-terrain objects are removed reliably while retaining most terrain details on generated DTMs.

I use a hybrid data structure that relies on the grid format. After filtering out the blunders with odd elevation or intensity values, the scattered 3-D points are transformed into grid-based range and intensity images by selecting the lowest point in each grid. Then, the HTRA is applied to generate DTMs using last-return range image. It applies the hierarchical strategy two times to estimate a high-quality terrain surface from lidar data with varying scene complexities. First, a rough DTM is obtained by generating the image pyramid bottom up with local minima selected. Second, the rough DTM is refined top down by identifying dense terrain points. Road points are identified by using multiple conditions, and vegetation points are discarded reliably using multi-return range data. The information about vegetation and roads can be used to adaptively adjust the parameters in the smooth condition so that one formula can be more suited to process different coverage and terrain types. The terrain elevation at a non-terrain point is interpolated using surrounding terrain points.

CHAPTER FOUR

ROAD EXTRACTION

Road extraction is one of the challenging tasks in digital photogrammetry and remote sensing. In this chapter, an approach is developed to automatically extract roads especially in urban areas using lidar data. Section 4.2 briefly describes the lidar dataset and related processing. The detection of road ribbons is discussed in Section 4.3. The reconstruction of grid road networks is discussed in Section 4.4. In Section 4.5, I evaluate the quality of results, and evaluate the performance of the method.

4.1 INTRODUCTION

Roads are critical components of the infrastructure. The correct and accurate information about their location, width and topology is important for various applications that support urban planning, traffic control, emergency response, and forest fragmentation models. The road information is usually stored in GIS, and is updated periodically, about every 1 to 5 years. Traditionally, road networks are digitized from ortho-photos and topographic maps manually or semi-automatically, or surveyed in site during engineering projects. This is time-consuming and expensive since extensive human labor is needed. The most used 2-D images also lack direct 3-D information. The increased demand for rapid and affordable production of road networks has triggered more research effort in exploiting new data sources especially when lidar and IfSAR data have lower pricing.

As reviewed in Sections 1.3 and 2.3, many algorithms have been developed for road extraction during the last three decades [4]. I address this task from two aspects:

- *The detection of roads using lidar data*. Researchers have investigated the extraction of roads using lidar data and/or ancillary imagery (Hofmann 2001; Song et al., 2002; Alharthy and Bethel, 2003). Roads are connected ribbons, which have certain width, are covered by homogeneous materials and are attached to the natural terrain. Roads

often have stable reflectance property and low heights in the DNM, are in general very smooth, and are separated by distinct patches in either reflectivity (e.g., vegetation) or height (e.g., buildings) relative to the bare surface (Hu and Tao, 2003).

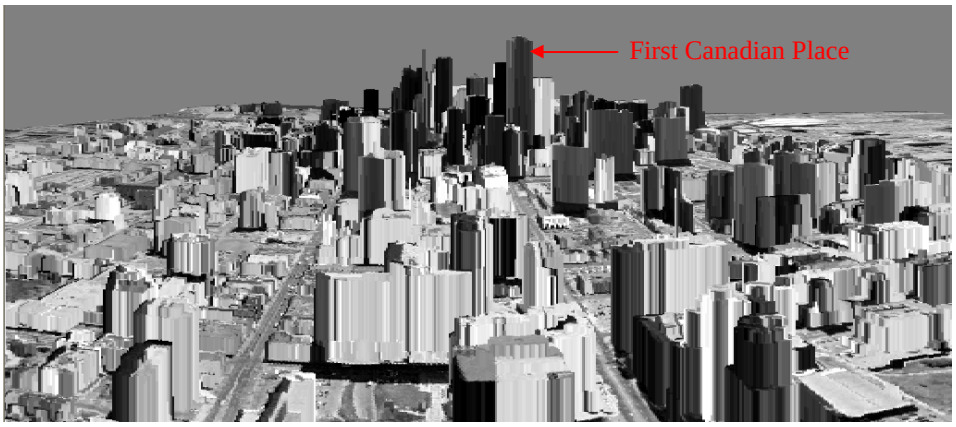
- Lidar range data can be used to improve the analysis of image data for detecting roads in urban areas and help detect the objects covered by shadows (Hofmann, 2001). But this method requires that the urban areas be relatively flat. This drawback can be eliminated by using DNMs in my algorithm (see Section 3.4.3).
- The relative separations between ground features (i.e., asphalt road, grass, building and tree) have been compared using intensity data. It is found that the separabilities are very high for road vs. grass and road vs. tree (Song et al., 2002).

• *The modeling of 3-D grid road networks* (GRNs). In many cities, road networks are arranged in a grid style in urban areas (Price, 1999; Hu and Tao, 2003). These GRNs are mainly composed of parallel and orthogonal straight roads with respect to the main orientation of the network. This simple geometry and topology constraint among grid streets may be used for road network reconstruction.

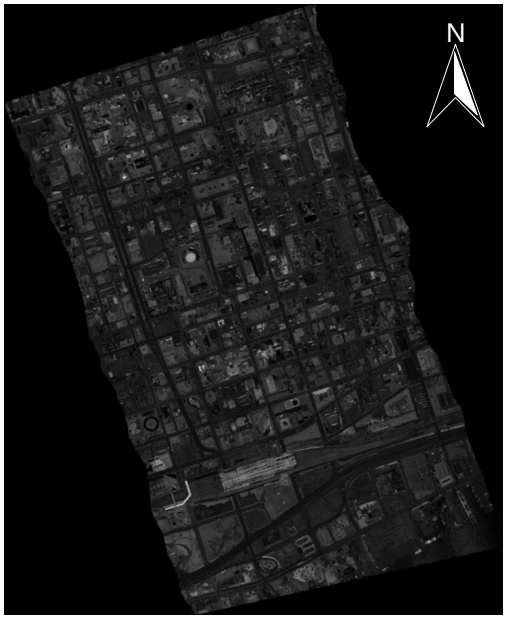
- Most algorithms extract 2-D free-form curvilinear roads, and are quite different in nature due to the differences in data sources, assumptions about road models and human interaction (Laptev, 1997, Park and Saleh, 2001).
- The 2-D roads detected using lidar data are represented as rough centerlines (Hu and Tao, 2003) or binary images (Alharthy and Bethel, 2003), which have very low quality and cannot be processed efficiently in GIS.
- A local grid constraint is applied to reconstruct 2-D street grids in Price (1999). But this method needs to manually select an initial seed intersection to provide the size and orientation of the GRN, and also assumes that road crossings be junctions of road segments of approximately constant width and height. My algorithm will achieve full automation by developing a global grid constraint.

In this chapter, I proposed automated approaches to detect road ribbons for complex scenes and reconstruct 3-D GRNs in built-up areas using multi-return intensity and range data. My method neither assumes a flat terrain, nor constant road width and height. First,

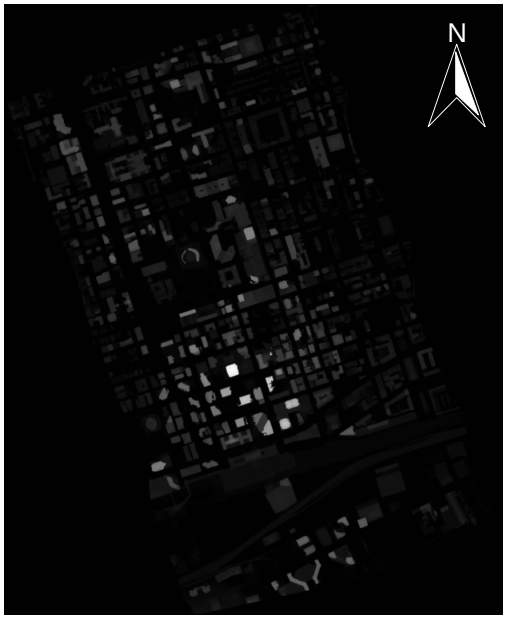
the range data is used to generate the DNM that provides height data. Then road ribbons are detected by a classification using intensity and height data. Finally the GRN is expressed by a 3-D grid road model (GRM) by detecting straight roads.



(a) The lidar DSM draped by an orthoimage (north to south looking)



(b) Intensity data



(c) Height data (the DNM)

Figure 4.1. Lidar dataset (Downtown Toronto)

4.2 DATA ACQUISITION AND PROCESSING

4.2.1 Study Area

This research focuses on downtown Toronto as a case study area for 3-D road extraction. In early 2002, Optech completed a flight mission of acquiring lidar data of Toronto urban

area using its ALTM 3020. The dataset provided is around the downtown region. Figure 4.1a shows an airscape of the lidar DSM draped by an orthoimage. The roads in the study area are coated with asphalt with pebbles or concrete. The first and last returns lidar range and intensity data were collected. The dataset consists of six strips each with about 1.75 million bi-return points and a density of 1.1 point/m^2. The overlap between two adjacent strips is about 40%. All the strips are combined into one data file containing about 10.6 million points. Table 4.1a lists the ranges of the raw elevation and intensity values.

A part of buildings and the complete road network were surveyed in field and delineated in 2-D. The Map Library at York University provided the DLGs and four aerial color DOQs with a 0.5-m GSD, which cover the study area. These photographs were flown in April 1999, and have a nominal scale of 1:20,000. Some road segments that are clearly discernible in the DOQs but not contained in the DLGs have been digitized manually.

Table 4.1. Information of the lidar dataset (Downtown Toronto)

		Number of points	**Laser return**	**Elevation range (meters)**	**Intensity range**
(a)	Raw data	10,666,043	Last-return	[-399.78, 398.41]	[1, 8160]
			First-return	[-399.72, 398.63]	[0, 8160]
(b)	Filtered data	10,632,257	Last-return	[20.04, 344.2]	[1, 288]
			First-return	[20.6, 350.7]	[0, 288]
(c)	Grid images	2811 x 2292	Last-return	[21.02, 343.51]	[2, 199]
			First-return	[21.2, 349.99]	[2, 199]

4.2.2 Lidar Data Pre-processing

The lidar data appears very noisy for both the intensity and the range data as revealed by their histograms. It is found that most points fall in significantly narrower intensity and elevation ranges than those counted for raw data. From the topographical map, I know that the minimum elevation should not be lower than 20-m ellipsoidal height in downtown Toronto.

In Figure 4.2, the odd points with too high (> 355 m) or low (< 20 m) elevations, or with too large intensity values (> 300) for either return of a pulse are identified, and then are analyzed in DOQs. The high points are all located on the roof of First Canadian Place [6] as annotated in Figure 4.1a. While the low points are located at a few spots beside which there are tall buildings. It is believed that these laser beams be reflected several times among the glasses of those buildings before they are detected, just like the multi-path effect of GPS. These specular reflections result in a longer travel time of the laser beam, and thus a lower elevation is calculated during post-flight processing. The large intensity points are distributed over the whole region mainly beside streets and in parking lots. This is due to the super reflectivity of power lines, metal structures of cars and constructions, or specular effects of glass. However, it is not sure which effect is most significant. These blunder points are screened out, including 781 high points, 268 low points and 32737 large intensity points. The new statistics for the filtered data are listed in Table 1b. Figure 3.12a shows a color rendering for the filtered dataset.

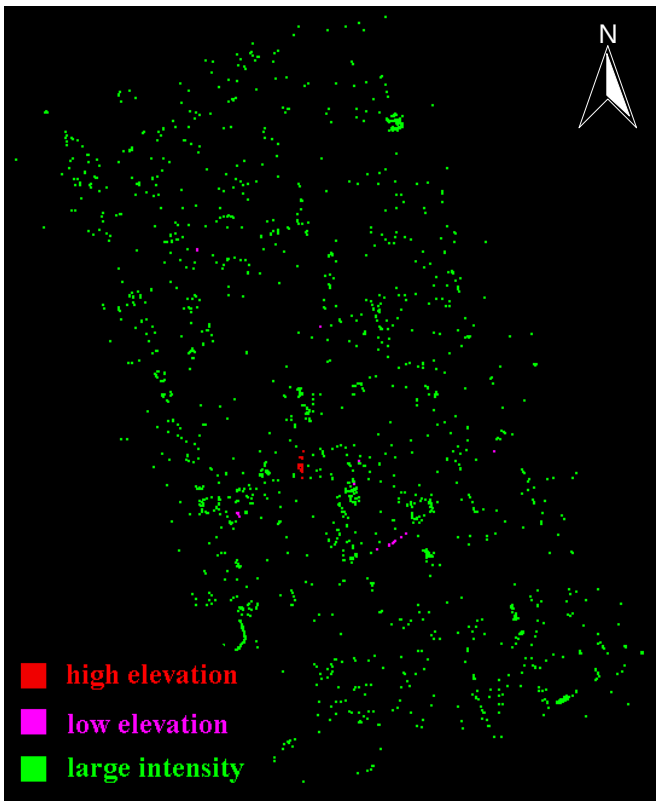


Figure 4.2. Blunder lidar points

4.2.3 Generation of the DTM, VSM and DNM

The first and last returns range and intensity data are used to generate the DTM using the HTRA. Specifying a 1-m GSD, intensity and range images are created spanning the study area. Some filters are used to further process these intensity and range images (see Section 3.2.2), and this deletes about 2,116 points with intensity values between 200 and 300. Figure 4.1b shows the intensity image for the filtered first-return data. The statistics for these range and intensity images are listed in Table 1c. Then the HTRA is applied to generate the DTM using last-return range data and also to obtain the height data (i.e., the DNM, see Figure 4.1c). The bi-return range data is also used to generate the VSM (see the green layer in Figure 4.3). Many tree points are removed rapidly using the VSM.

4.3 DETECTION OF ROAD RIBBONS

In reflectivity, the spectral signature of asphalt roads significantly differs from vegetation and most construction materials. The reflectivity rate of asphalt with pebbles is 17% for the infrared laser as indicated in Table A.2, and no other major materials have a close reflectivity rate. In height, pavements are attached to the bare surface and appear as smooth ribbons separating the street blocks in a city. Thus roads can be roughly described by cross-connected ribbons that are of constant reflectance and very low height. To this end, I apply to intensity and height data a classification to detect road ribbons and road centerline models (RCMs) in urban areas.

4.3.1 Road Ribbons Detection Using Intensity Data

The intensity data is usually very noisy, and the moving objects (e.g., cars) and (white or yellow) lane markings on roads also result in intensity speckles. In intensity image, the asphalt roads appear as dark ribbons. Besides asphalt, concrete is another paving material. Concrete and asphalt have different but very close reflectivity (see Table A.2). In ideal condition, it is possible to distinguish one from another, and thus the road class would have two clusters. But, if these materials are used in roads, due to wearing of vehicle tires, black colored rubber particles begin to stick on their surface after the road become operational. As a result, concrete road surface gives similar kind of reflectance as asphalt

(Hazarika et al., 1999). This is also true for infrared laser as proved by the classification results (Song et al., 2002). I first depress the negative effect of noise using the adaptive smoothing algorithm (see Section B.1.5).

If a point passes the test formulation in a decision function for the road class, then it will be a road pixel. To extract asphalt roads, the intensity image is segmented using dual thresholds enclosing the intensity range of the road class. A membership function is needed to calculate the degree to which a pixel has to meet to be classified as the road class, and is defined as

$$RC_I(p) = \begin{cases} 1 & Ti_l \le I(p) < Ti_h \\ \exp((I(p) - Ti_l)/Si) & Ti_l - Si \le I(p) < Ti_l \\ \exp((Ti_h - I(p))/Si) & Ti_h \le I(p) < Ti_h + Si \\ 0 & otherwise \end{cases} \tag{4.1}$$

where Ti_l and Ti_h are the low and high thresholds, respectively; $I(p)$ is the intensity value of the point *p*; *Si* is the intensity scale (e.g., (Ti_h - Ti_l)/2). Since the reflectivity of asphalt is fixed, a base threshold *Ti* can be determined from a priori reflectivity rate or in a supervised manner by selecting some road pixels in the intensity image. Then histogram analysis is used to calculate an allowable intensity range so that each threshold is the closest local minimum at either side of the base threshold (Papamarkos and Gatos, 1994).

In Figure 4.3, the segmented results of the intensity image are displayed in the red layer (last-return) and the blue layer (first-return) with the VSM placed in the green layer. The low and high thresholds are 10.4 and 23.7, respectively. The magenta pixels indicate the presence of roads, and black pixels represent the background. I also tried the ISODATA classification in ERDAS IMAGINE 8.5 Classifier to verify this segmentation. The resulting intensity range for the pavements is [11.3, 21.6] [9]. That is, the thresholding approach and the ISODATA classification get very close segmentation results.

[9] In the unsupervised ISODATA classification, the number of classes is 10, the number of iterations is 8, the convergence threshold is 0.95, and the standard deviations are determined automatically. Then the generated signatures are analyzed with respect to selected road pixels to obtain the thresholds.

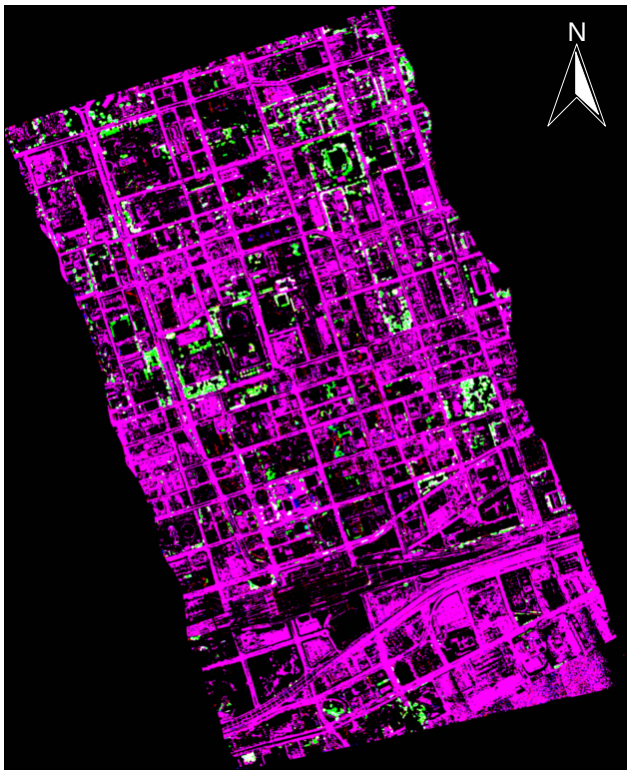
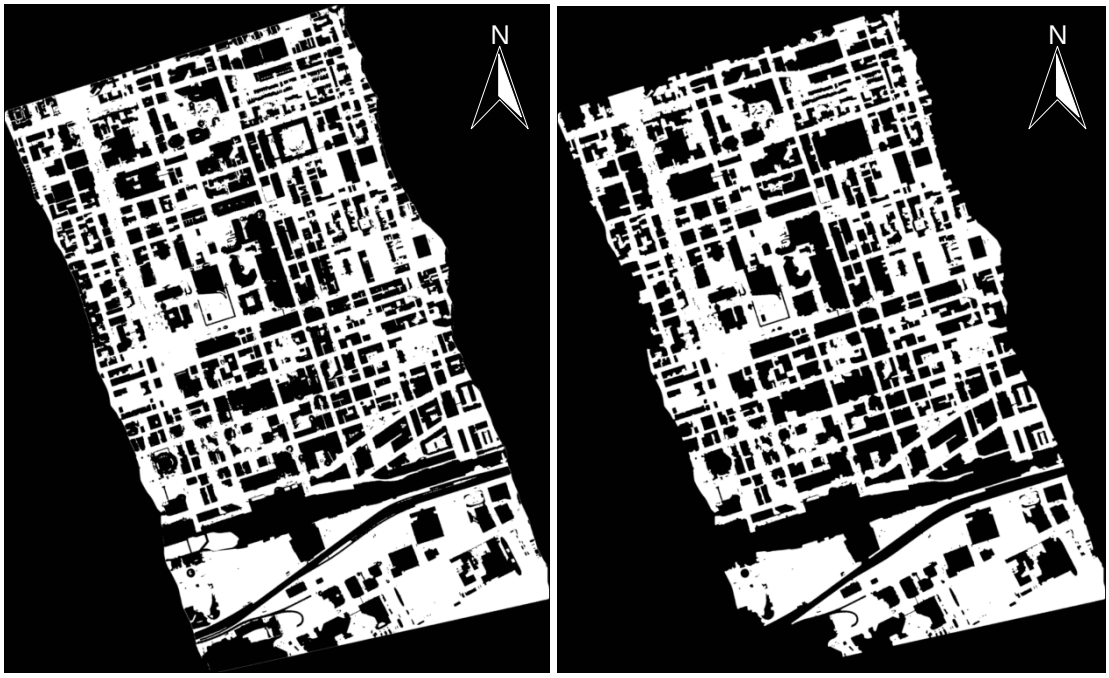


Figure 4.3. Road ribbons by classifying intensity data

As illustrated in Figure 4.3, the last return may hit on the ground in forested areas and thus mis-classify some vegetation into pavements, while the first-return can correctly classify the vegetation. So the first-return intensity data is more suitable for the classification in forested areas. But the misclassification of the last return may be eliminated with the aid of the VSM. The small black holes surrounded by roads are usually trees in the streets. At the bottom, a part of water bodies are wrongly classified as roads. The misclassification in other areas is neglectable and also does not influence the detection of terrain points because those points in water areas represent close terrain elevations slightly smaller the bank. Because road ribbons form a network of connected components, the isolated building roofs covered by the same material will be removed rapidly when performing connected component analysis (see Section B.1.7) and deleting those objects whose areas are smaller than a large threshold (say 10000 m^2). Although not all the pavements are detected and some misclassifications occur, the result is sufficient for serving as auxiliary information for detecting candidate road points.



(a) DNM Segmentation (b) Isolated objects removed

Figure 4.4. Road ribbons by classifying the height data

4.3.2 Road Ribbons Detection Using Height Data

It is a fact that roads are thin pavements over the terrain surface. Once the DNM is created, I can obtain a rough model for road ribbons by detecting those areas with low DNM heights. The membership function for the road class is defined as

$$RC_H(p) = \begin{cases} 1 & H(p) < Th \\ \exp((H(p) - Th)/Sh) & Th \le H(p) < Th + Sh \\ 0 & H(p) \ge Th + Sh \end{cases} \qquad (4.2)$$

where Th is the height threshold; $H(p)$ is value of the point p in the DNM; Sh is the height scale (e.g., 3 m). Figure 4.4a shows the segmented result with Th equal to 1.8 m that is calculated by an iterative selection method (see Section B.1.6). Figure 4.4b shows a result with isolated objects removed by a connected component analysis, and the area threshold is 20000 pixels. The road ribbons obtained this way are of unsatisfactory quality because open areas including grasslands are also included. So the results may be used as auxiliary information to refine the intensity based classification results.

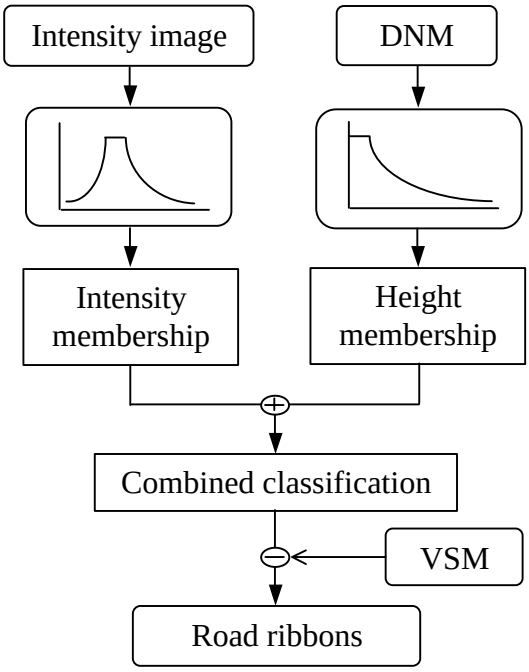


Figure 4.5. Flowchart of the road detection algorithm

4.3.3 Road Ribbons Detection Integrating Intensity and Height Data

It can be easily found that integrating intensity and height data may produce more reliable road detection results. The workflow of the road detection procedure is illustrated in Figure 4.5. On the one hand, the intensity provides the spectral reflectivity, which can help identify most roads even if the objects coated by the same material are also included. On the other hand, the height data can help identify most non-building and non-forest areas even if those low open areas such as grasslands are also included. Using height information, the built-up areas with higher elevations than their surroundings will be safely removed; while using the (first-return) intensity information, the vegetated areas are easily removed. So it is possible to separate roads from trees, buildings and grasslands with minimum misclassification when fusing the intensity and height data. In detail, compared to roads, grasslands have different intensity although they have low elevation, trees have different values in both intensity and height, and buildings have high structures with elevation jumps although they may be coated rainproof asphalt.

The membership function for integrated road classification is a weighted sum of two

components given in Equations 4.1 and 4.2, and is defined as

$$RC_{IH}(p) = RC_I(p) \cdot w_i + RC_H(p) \cdot w_h, \quad w.r.t. \quad w_i + w_h = 1 \tag{4.3}$$

where w_i and w_h are weights for intensity and height information, respectively. However, parking lots are still kept because of same reflectance and low heights as roads, and bridges and viaducts are removed because of their large heights. In Figure 4.6, the classification results are placed in the red layer (last-return) and blue layer (first-return) with the VSM in the green layer. It is observed that only neglectable differences exist because the tall vegetation areas from classifying the last-return intensity have been removed using the height information from the DNM.

Figure 4.6. Road ribbons by classifying both intensity and height data

The segmented results are further processed to produce a clean shape. First, the initial road ribbons are refined by subtracting the VSM (see Section 3.5) if the classification is

performed on the last-return intensity, and this step is skipped when using the first-return intensity. Next, the objects not connected to the road ribbons are deleted. This is to label connected components in the road image and to clean small objects (e.g., < 10000 pixels). Finally, the morphological opening and closing operations are applied on the image. The effects of these operations are to isolate the buildings weakly connected to road ribbons and to fill small void spots in the middle of roads due to the disturbance of trees. Again, the isolated blocks are removed by connected component analysis.

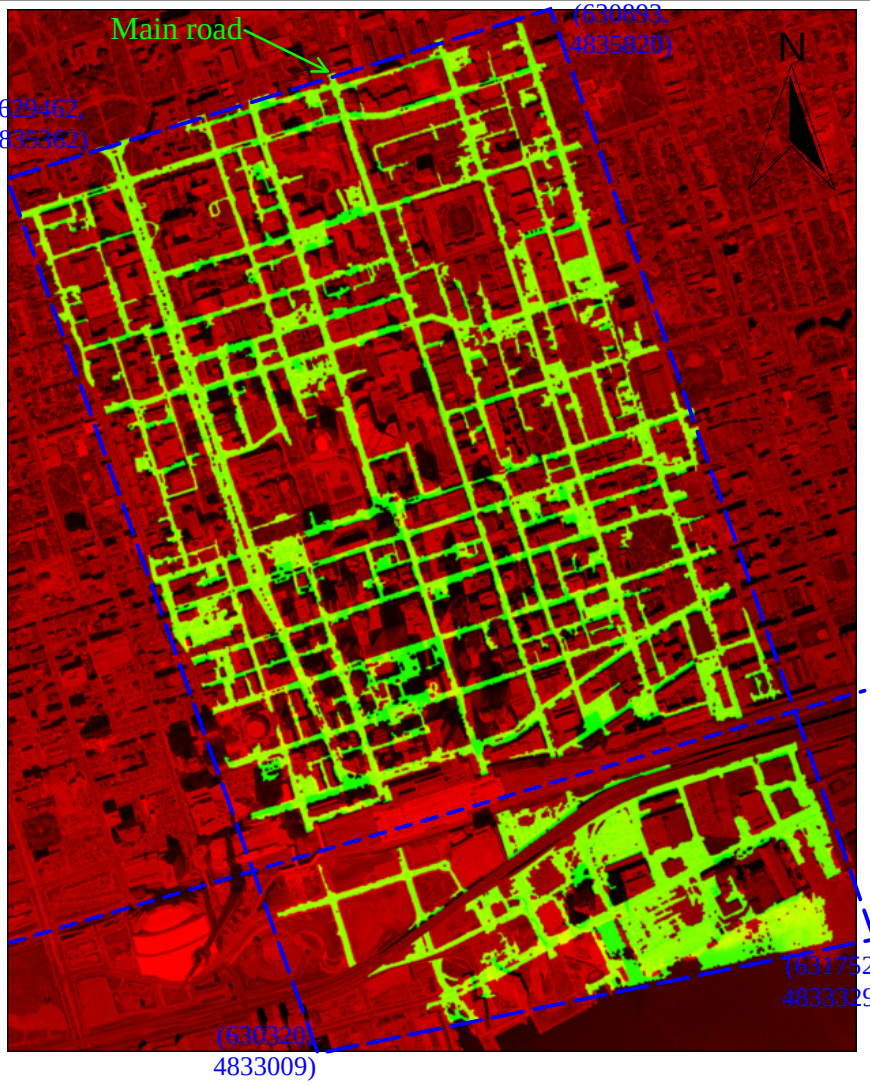


Figure 4.7. Final road ribbons overlaid on the DOQ

As shown in Figure 4.7, the final roads (green layer) match the road structures in the DOQ (red layer) well at the upper part since the ribbons are correctly overlaid on most

road stripes. But at the lower part, some water regions are still wrongly classified as roads, and outdoor parking lots connected with the roads are kept yet. More results about the detection of road ribbons are illustrated in Figure 3.7 (Village site) and Figure C.1 (Santa Barbara airport). Extra information can be derived about the study area from these road ribbons. For example, the area covered by the road class (including roads and parking lots) is 1.05 km^2, which is 27.9% of the study area (about 3.76 km^2) enclosed by the dashed quadrangle in Figure 4.7. This indicates a dense road construction in downtown.

(a) Thinned road ribbons (b) Highlighted nodes

Figure 4.8. Rough road centerlines

Table 4.2. Number of road pixels during road classification

Figure	4.3	4.4b	4.6	4.7	4.8a
Number	1662220	2058315	1195092	1050172	84601

4.3.4 Rough Road Network Generation

The road ribbons give a representation about the layout of pavements, and cannot well delineate the road network. The results also cannot be efficiently processed in GIS. A rough description of road centerlines (see Figure 4.8a) is obtained by thinning (see

Section B.1.8) these road ribbons. This road centerline model (RCM) will reduce computational burden and avoid confusion afterwards. The road crossings having more than three 8-neighbors are highlighted in Figures 4.8b. However, such a RCM is geometrically inaccurate and cartographically unusable. Table 4.2 shows the number of road pixels obtained during the road classification procedure.

4.4 RECONSTRUCTION OF GRID ROAD NETWORKS

To reconstruct grid streets automatically from the RCM, a novel procedure has been developed. First, a rough GRM composing of all the possible road primitives (i.e., road segments and crossings) is generated by applying the global grid constraint. Then, a verification processing is deployed to validate segments and form the final GRM.

4.4.1 Sequential Hough Transformation

After image pixels are transformed to Hough space using the 2-D HT (see Section B.1.12), lines can be found by detecting peaks in the parameter domain. The coordinates (x, y) of the intersection of two non-parallel lines can be calculated by

$$x = \frac{r_1 \sin q_2 - r_2 \sin q_1}{\sin(q_2 - q_1)}$$
$$y = \frac{r_2 \cos q_1 - r_1 \cos q_2}{\sin(q_2 - q_1)} \qquad (4.5)$$

where (r_1, q_1) and (r_2, q_2) are parameters of the two Hough lines. In conventional Hough transformation, the search of the second maximal position is problematic since we do not know how to perfectly remove the neighbors of the first peak in Hough space. Zhang and Burkhardt (2000) proposed a sequential grouping method to form lines from edge points. The main idea is to find and delete the lines one by one. First, from the position of the global peak, the points belong to that line can be grouped. Then those points that are grouped from this peak are deleted in both Hough space and image space. This inverse Hough operation eliminates the contribution of the points belonging to the

found line; thus we can reliably find the next global peak from remaining points. This processing is repeated until all the lines are detected. This sequential HT (SHT) may be enhanced by processing oriented pixels instead of pixel positions only to allow for more rigorous grouping for complex pattern recognition as discussed in the next chapter.

Table 4.3. Information about the grid road models

Primitive type	**Properties**	(a) Rough GRM	(b) Validated GRM
Road segments	Total number	1214	402
	Width range (m)	[0, 52.7]	[3.9, 52.7]
	Average width (m)	8.72	16.11
	Length range (m)	[0.4, 203.9]	[11.4, 203.9]
	Average / total length (m)	74.7 / 90726	118.4 / 47602
	Maximal slope (degrees)	14.25	8.28
Road crossings	Total number	649	302
	Elevation range (m)	[21.1, 67.7]	[36.2, 64.5]

4.4.2 Procedure for Reconstructing Grid Roads

As can be observed from Figure 4.7, the road network is composed of cross-connected long streets in grid arrangement except the bottom part below the green dashed line, which is kept in the processing to test the robustness of the algorithm. When transforming the road centerlines to the Hough space, there will present a number of peaks mainly distributed inside two slim zones that are shifted by 90^o in the Hough accumulator. Based on this fact, the GRM extraction procedure is divided into four steps:

1. Find out all the main road lines guided by the global grid constraint. A constrained searching in Hough space is developed using the SHT.
   1.1. Detect the main street. The global peak (r_0, q_0) indicates the main road of the whole network, and it accumulates to 2288 in Hough space. As pointed out by an arrow in Figure 4.7, the main angle q_0 is 18^o, and thus the main

orientation of the whole network is 108^{o}.

1.2. Detect all the lines parallel to the main street. This is to search local peaks within a slim buffer along the main angle.

1.3. Detect all the lines perpendicular to the main street if the layout of the road network likes a chessboard. This will search local peaks within a buffer along the angle of $q_0 + 90^o$ or $q_0 - 90^o$, whichever is between 0^{o} and 180^{o}.

1.4. Detect tilt roads that are neither parallel nor perpendicular to the main orientation by detecting remaining peaks.

The street details are controlled by several thresholds, including the minimum road width, minimum road length and maximum angle tolerance. For example, more road segments will be detected if a small width threshold is used.

2. Produce a rough GRM, which is expressed as connected road segments.

2.1.The grid crossings are calculated by intersecting all the parallel lines with all the orthogonal lines via Equation 4.5, and road segments connecting each pair of immediately adjacent crossings are created.

2.2. The tilt lines are also intersected with main lines having a convergence angle larger than 45^{o}, and tilt road segments are created by connecting each pair of adjacent crossings on the tilt lines.

2.3. Each road crossing is a node in the GRM, and is assigned with a counter that is the number of segments connected to it.

Figure 4.9a shows such a GRM (red layer) generated using the road centerlines (white layer). The GRM is of good quality even if the bottom part does not shape well and adds some in-determinacy to the algorithm. Table 4.3a lists some statistics of the model.

3. Validate the GRM. The road crossings and segments in the rough GRM are overlaid on the binary road ribbons. Each segment is verified separately.

3.1. Early skipping. If the length of the segment is smaller than a threshold (e.g., 10 m), then reduce the counters of its two associated nodes and jump to process the next one.

3.2. Interpolate multiple (e.g., 3 to 9) vertices between the two nodes depending on its length, and verify if each node or inter-median vertices is valid. A

vertex is said valid if it has enough neighboring road pixels.

- If most vertices are invalid, then this segment is discarded, the counters of both associated nodes are reduce by one, and jump to process next one.
- If more than a half of vertices are valid, then it is labeled as a candidate road segment.

3.3. Compute width for each candidate road segment.

- At each inter-median vertex, a width is calculated by analyzing the 01 and 10 patterns along the perpendicular direction.
- The overall width is calculated as the alpha-trimmed mean of all the widths at those inter-median vertices.
- A segment is discarded if its width is less than a threshold (e.g., 3 m).

Figure 4.10 shows road segments for four cases (green), and their properties including width, length and slope. If a segment displaces the road centerline, then it is refined by a simple translation.

3.4. Check the connectivity topology of road crossings. The counter of a node indicates its connectivity strength, and is processed as below.

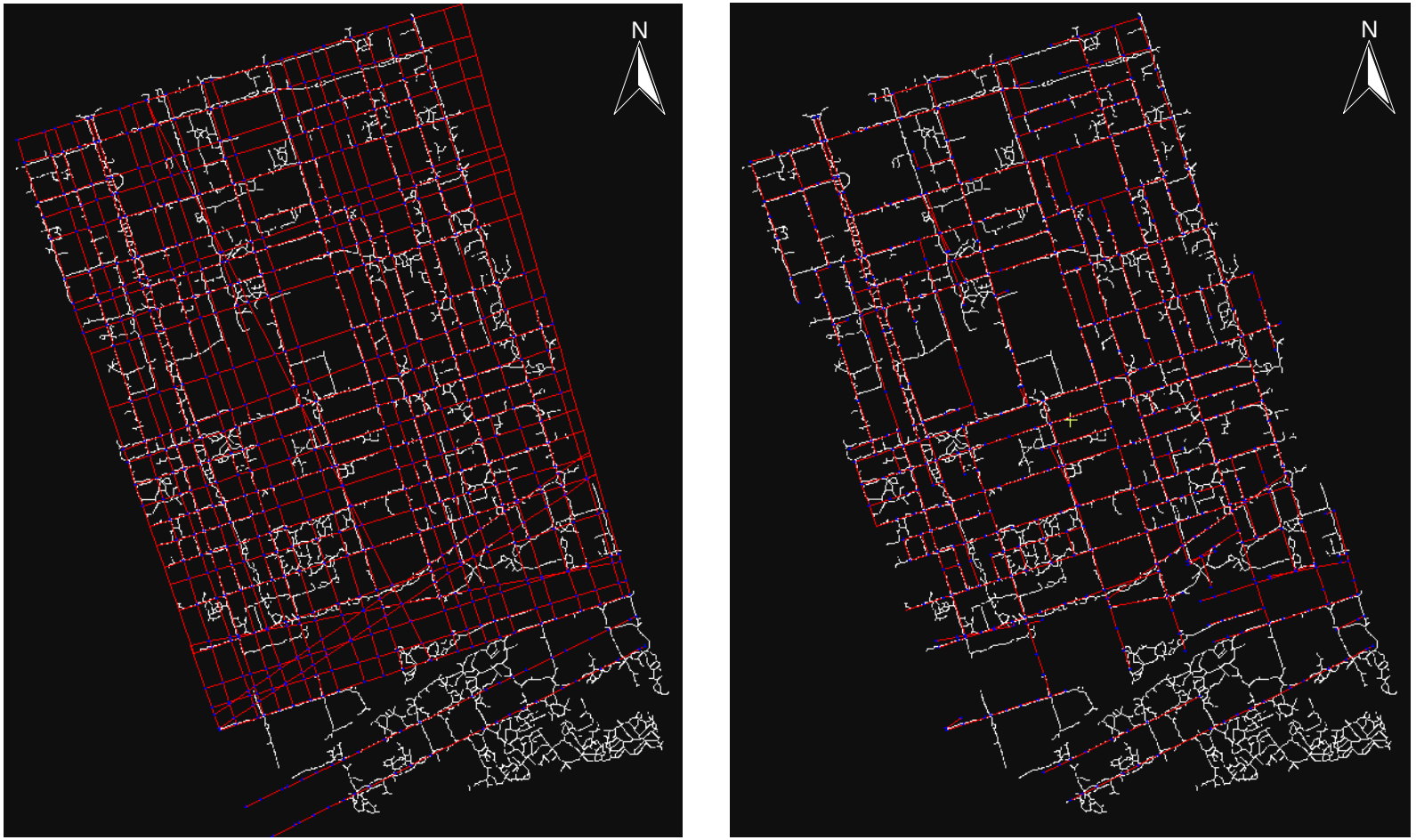
- If the counter of a node has been reduced to zero, then delete it.
- If the counter of a node is equal to 2 and the two associated segments have same orientation, then merge the two segments and delete the node, or convert the node to an inter-mediate vertex of the merged segment.

In Figure 4.9a, the numbers of the first class and the second class of nodes is 128 and 219, respectively. The validated GRM (red layer) is shown in Figure 4.9b, and it fits those centerlines (white layer) well in most cases. Table 4.3b lists some statistics about the validated GRM.

4. Output 3-D road network. The image coordinates of the nodes are transformed to coordinates in the UTM map projection coordinate system.

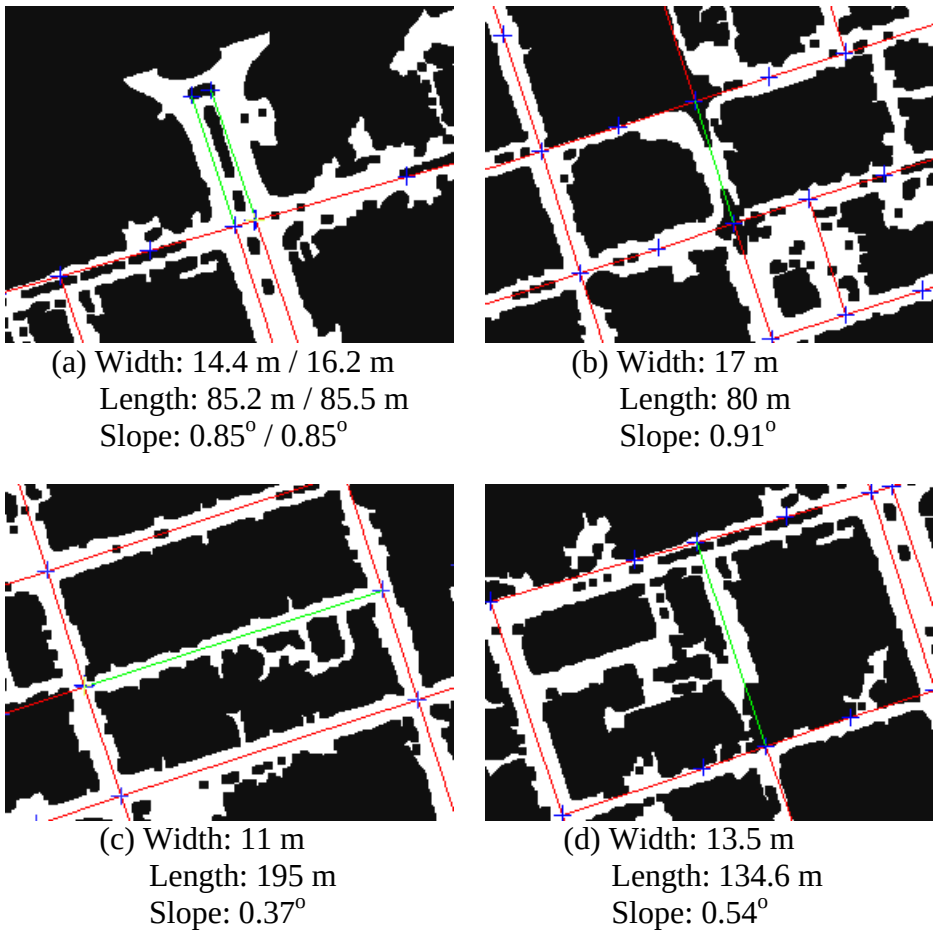
4.1. Retrieve elevation for each node of the validated GRM from the derived DTM, and discard those falling in void areas of the lidar data.

4.2. Compute the road slope. A segment becomes a polyline when inserting inter-mediate vertices at places where the slope changes significantly.



(a) The rough GRM (b) The validated GRM

Figure 4.9. The grid roads overlaid on the road centerlines



(a) Width: 14.4 m / 16.2 m
Length: 85.2 m / 85.5 m
Slope: 0.85° / 0.85°

(b) Width: 17 m
Length: 80 m
Slope: 0.91°

(c) Width: 11 m
Length: 195 m
Slope: 0.37°

(d) Width: 13.5 m
Length: 134.6 m
Slope: 0.54°

Figure 4.10. Four cases for road segments

4.4.3 3-D Reconstruction Results

Assuming a maximum road width of 60 m, an instance of the 3-D GRN of downtown Toronto is shown in Figure 4.11, where the road segments (blue lines) and road crossings (red crosses) are overlaid on the DOQ draped on the derived DTM. The roads are plotted according to their relative widths. The DTM is displayed at 50% transparency. The street blocks reconstructed using the sequential linking technique (see Section 5.6) are also depicted. In Figure 4.12, a part of another extraction instance assuming a maximum road width of 40 m is compared with the surveyed street elements in the DLGs. The extracted road crossings and segments are rendered in green and red, respectively. The surveyed road crossings and segments are rendered in cyan and blue, respectively. Compared with the surveyed vector data, the extraction results show the reliability of the method for reconstructing 3-D GRNs.

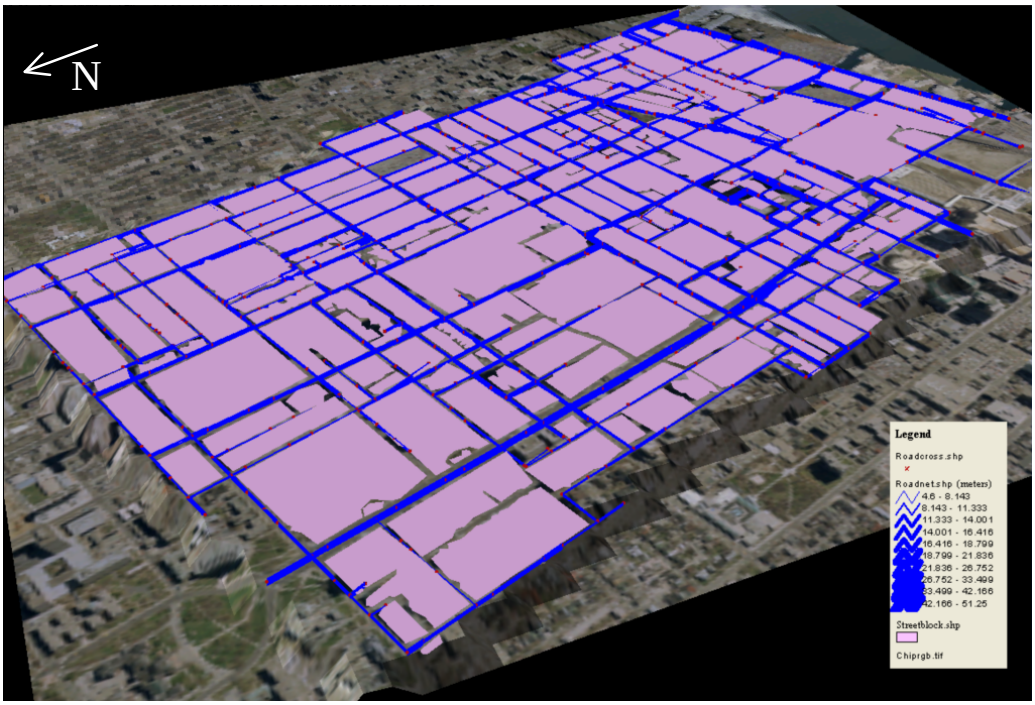


Figure 4.11. 3-D road network overlaid on the DOQ draped on the DTM

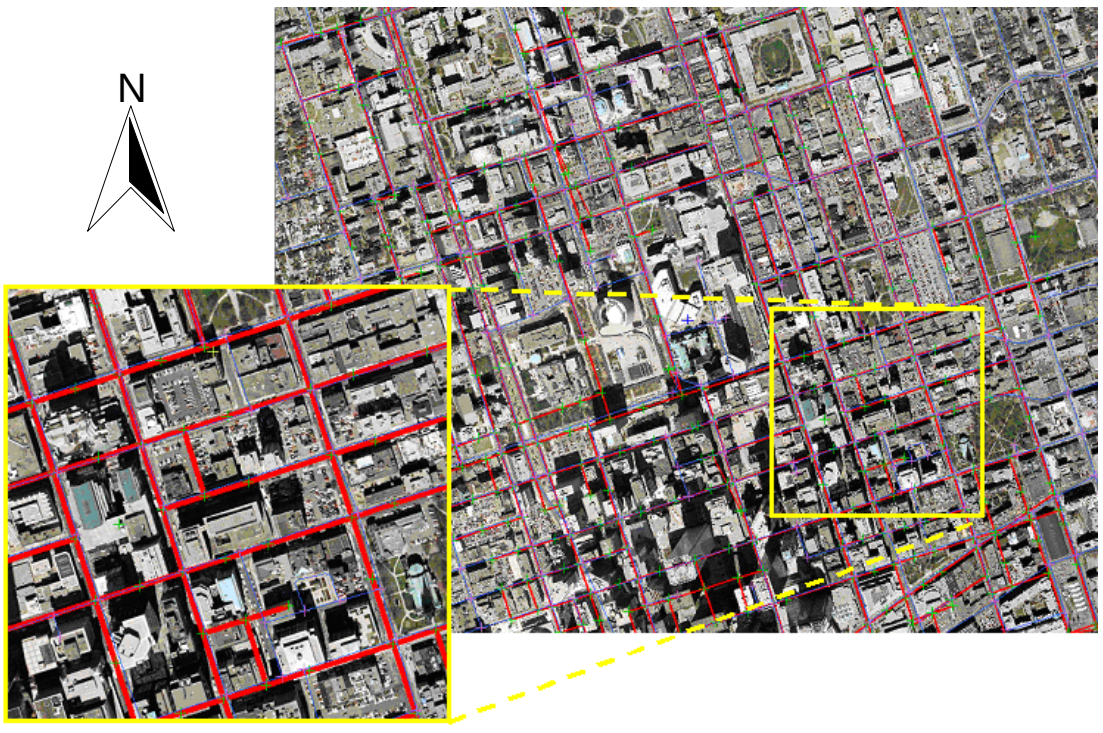


Figure 4.12. The extracted roads and the reference map

4.5 RESULTS EVALUATION

The quality of the extracted roads and the performance of the proposed algorithms are evaluated for the two major works:

The detection of road ribbons

This algorithm has been tested using datasets acquired at residential (Village site), urban (downtown Toronto), mountainous, suburban and forested (Santa Barbara airport) areas.

- The integration of both the radiometry property of intensity data and the height information of the DNM is highly feasible in reducing the misclassification of the road class as shown in Figures 3.7, 4.7 and C.1.
  - The road ribbons match the actual roads well as found by overlaying them in DOQs or lidar images and comparing them with corresponding road patterns.
  - The first-return intensity data is more suited to characterize the road class to avoid the misclassification of some vegetated areas into roads since the last return may hit on the ground under trees (see Figure 4.3).
  - The classification results using the last-return intensity data can be refined by subtracting the VSM derived from multi-return range data, while the classification results using the first-return intensity data do not need this refinement since vegetation has different reflection rates than asphalt (see Figure 4.3).

- The resulting road ribbons are not perfectly complete due to some classification errors (see Figures 3.7, 4.7 and C.1).
  - Some roads are newly paved with concrete that has a larger although close reflection rate than asphalt. This can be solved by human interaction to determine a slightly wider intensity range for the road class if the road situations are known.
  - The bridges and viaducts are not detected because they have large heights. This needs ancillary data to solve this problem.
  - In the situation where the roads are coated with materials other than asphalt, the

reflection properties of those materials in the intensity data should be analyzed in the same way as done for asphalt.

- The resulting road ribbons contain non-road areas, including outdoor parking lots and some water areas (see Figures 4.7 and C.1). The reason is that they give similar reflection rate as asphalt and nearly zero height as roads. This need ancillary data, and is also difficult to be solved using lidar data only.

- My results are very close to those given in Alharthy and Bethel (2003) by a visual comparison since very similar procedures have been developed for road classification using multi-return intensity and range data. Alharthy and Bethel's (2003) road detection algorithm works well with a limited set of characteristics, and may fail when these characteristics change beyond the expected limits. For example, it might fail in areas where the road materials vary. This is also true for my algorithm as discussed above.

The reconstruction of 3-D GRNs

This algorithm has been tested for downtown Toronto. The road primitives in the validated GRM are compared with surveyed street elements used as the reference model. Instead of matching the primitives included in the reference map and those in the GRM automatically as proposed by Wiedemann (2002), I chose to count the matches manually to avoid introducing new uncertainties. A road segment is matched if its centerline is falling in the closest road stripe in the DOQ. A road crossing is matched if its connectivity relations are same in the GRM and the reference map, and it is falling in the closest road intersection in the DOQ.

Table 4.4 shows the evaluation results for the upper part of the GRM (see Figure 4.7), which has 391 road segments and 285 road crossings in total. The average length of these road segments is 115.7 m. The bottom part is excluded from evaluation because it is kept mainly for robustness testing as claimed before. The definition of the quality measures

used in Table 4.4 can be found in Section B.3.

Table 4.4. System performance evaluation

		Road segments	**Road crossings**
Quantities	TP	358	223
	FP	33	62
	FN	68	46
Quality measures (%)	Completeness	84	83
	Correctness	91.6	78

- The GRM matches the actual GRN well in urban areas with regular street grids.
  - The road centerlines obtained by thinning road ribbons give a rough but clean description of the actual road centerlines although they geometrically inaccurate and cartographically unusable (see Figures 4.8 and 4.9). These rough road centerlines reduce the computational burden in Hough space and greatly eliminate the confusing when detecting peaks.
  - The road segments and crossings have the completeness of 84% and 83%, respectively, and the correctness of 91.6% and 78%, respectively.

- The road segment based hypothesis and verification strategy leads to efficient, robust and automatic reconstruction of 3-D GRMs.
  - The algorithm can reconstruct GRNs robustly even if the grid constraint is not fully satisfied. That is, some parts of the road network may be in irregular patterns (see the bottom part in Figures 4.7 and 4.9b).
  - Besides grid roads, 8 main tilt roads are correctly detected (see Figure 4.9b). This again shows that the algorithm is robust to the non-grid roads to some extent.
  - The resulting GRMs hold topology information of the road network that is useful for network analysis in GIS (see Figures 4.9b and 4.11).
  - The algorithm is limited to process a special case because of the use of the global grid constraint. Therefore, a premise of the use of this algorithm is the automatic determination of grid areas.

- For road segments, the average lengths are 118.7 m, 82.6 m, 78.7 m, respectively for TPs, FPs and FNs. This shows that the redundant and missed road segments are relatively shorter than the true positives.

- For true positive road segments, their three properties are calculated.
  - The length differences in both the GRM and the reference map are used to compute two statistics: 0.95-m mean error and 1.3-m RMSE.
  - The visual checking shows that the calculated road widths are proportional to the road patterns in DOQs (see Figure 4.12).
  - The road slopes are not checked because the surveyed streets give only 2-D road centerlines, and thus no slopes can be calculated for segments in the reference data.

- For false positive road segments, most are grid roads and occur at some short tilt roads where the grid constraint is not well satisfied; and remainings are tilt roads wrongly introduced when detecting peaks of tilt roads in Hough space but they are not removed utilizing information contained in road ribbons (see Figure 4.9b and 4.12). If the thresholds for detecting these peaks are increased to avoid introducing corresponding FPs, then more FNs will occur.

- For false negative road segments, most are grid or tilt roads that are relatively short (see Figure 4.12), and their peaks are unlikely detected reliably in Hough space. If the thresholds for detecting peaks are decreased to find those short grid and tilt roads, then more FPs will occur. That is, the algorithm needs a good trade-off in setting appropriate thresholds to minimize FPs and FNs.

- For true positive road crossings, the 2-D positions have a 0.9 m RMSE by comparing the GRM and the reference model. This shows that these road crossings are located correctly and accurately since the real road intersections have diameters not less than 10 m.

- My algorithm can output 3-D street grids, while Price (1999) demonstrated the extracted 2-D street grids for a district of Washington DC. The road segments are often misplaced

by the road width in Price (1999). While this problem happens in a few conditions in my results, the displacements are removed during computing the width attribute.

4.6 SUMMARY

I developed new algorithms for automated detection of road ribbons for complex scenes and reconstruction of 3-D GRNs in urban areas using multi-return intensity and range data. The detected road ribbons match the actual roads well in most areas as found by comparing against the road textures in DOQs. The comparisons between the extracted road primitives and the surveyed streets show the reliability of the method for reconstructing 3-D GRNs and that the GRMs match the GRNs well in urban areas.

Integrating both the radiometry property of intensity data and the height information of range data reduces the misclassification of the road class, and is highly feasible for the detection of road ribbons. Rough road centerlines are obtained by thinning detected road ribbons. This algorithm has been tested using several lidar datasets acquired at urban, mountainous or forested areas and obtained good results. In urban areas where the road networks are arranged in grid style, a new method is proposed to utilize global geometry and topology constraints to reliably reconstruct GRNs. The basic idea is to use the grid constraint to formulate hypothesis for the presence of road primitives and to verify this by checking multiple criteria. The rough centerlines can be refined by detecting straight lines using the SHT and calculating intersections among lines. The attribute values of road segments including length and width should not violate their a priori ranges, and are used as evidences indicating the presence of roads. The presence of a road crossing relies on the evidence about the presence of road segments, one of whose end nodes is this crossing. The vertical coordinates of road crossings are retrieved from the lidar-derived DTM. The major attributes about each road segment including its width, length, and slope are calculated. The produced GRM is composed of 3-D road segments and road crossings hold the connectivity topology.

CHAPTER FIVE

BUILDING EXTRACTION

Building extraction is one of the main research topics of the photogrammetry community. This chapter presents algorithms for building extraction from lidar data. The workflow of the methodology is briefly described in Section 5.2. Sections 5.3 to 5.8 discuss the proposed algorithms in detail. In Section 5.9, I show results using three datasets, evaluate the quality of results, and discuss the performance of the algorithms.

5.1 INTRODUCTION

Automatic extraction of man-made structures such as buildings from geospatial data is of great practical interest for a number of applications. Sample applications include urban planning, environmental studies, wireless communication, marketing, and change detection. The primary data sources used for building extraction are becoming increasingly available from a variety of platforms and sensors, such as aerial and satellite images, radar images, IfSAR data, and lidar data.

As reviewed in Sections 1.3 and 2.4, many algorithms have been developed for semi-automatic or automatic extraction of buildings using lidar data in the last decade [4]. The building models used in GIS are mainly produced by semi-automatic approaches (Rottensteiner, 2001). The task of automated building extraction is difficult due to many reasons. I address this task based on its two sub-procedures:

- *Building detection*. Building detection determines the locations of building footprints, which are used by subsequent reconstruction (Weidner, 1995). This step should distinguish buildings from vegetated regions. To date, general building footprints cannot be reliably detected automatically.
  - Buildings may be detected by slicing range data and analysing the hierarchy of consecutive object segments (Zhan et al., 2002a; Baltsavias et al., 1995).) The

range data can be used to improve the analysis of image data for detecting buildings in urban areas and help detect the objects covered by shadows (Hofmann, 2001). But this method requires that the urban areas be relatively flat. To avoid the disturbance of terrain relief in a large scene, some researchers have used height information obtained by subtracting the lidar-derived DTM from lidar DSM (Weidner and Forstner, 1995; Haala and Brenner, 1999; Gamba and Houshmand, 2002; Rottensteiner and Briese, 2002; Hu et al., 2003). But most of this type of algorithms used simple DTM generation algorithms and thus obtained in-accurate height data. This problem has been well solved by developing the HTRA (see Chapter 3) to generate accurate DTMs for complex scenes.

- The reliable reconstruction of complex footprint boundaries is a key step, but is seldom discussed in detail. Most algorithms works well only under specific assumptions, which limit footprints to simple shapes such as rectangles or low-quality polygons (Weidner, 1995; Vosselman, 1999; Wang and Schenk, 2000). Other algorithms, which do not make such assumptions, often got distorted boundaries expressed by edges detected from lidar DSMs or DNMs (Baltsavias, 1995; Weidner, 1995; Yoon et al., 1999; Wang and Schenk, 2000; Rottensteiner and Briese, 2002). These boundaries need to be refined using a set of geometric regularity constraints (Vestri and Devernay, 2001). At present, there are no solid algorithms available at this processing stage. I will improve this situation by developing robust and efficient algorithms based on Hough transformation.
- To distinguish building footprints from vegetated regions, the classification is often based on shape measures assuming some geometric regularity constraints (Wang and Schenk, 2000) or the roughness of the point clouds. These measures limit the detectable buildings to a narrower spectrum, and also are not very reliable for complex scenes such as densely forested areas. The shape measures often make use of 2-D properties such as area and perimeter; while complex building roofs may present close values when calculating the roughness measures. The use of multi-return range data can benefit the separation of buildings and vegetation since building roofs must be solid surface (Zhan et al., 2002b; Hu et al., 2003). Lidar cannot penetrate solid surfaces and will get a single return only for

them. That is, the first and last returns are same in elevation at solid surfaces but are different at vegetated regions. However, lidar gets the similar effect at building boundaries as that at vegetated areas. This problem has been well solved in Section 3.5.2.

- *Building reconstruction*. Building reconstruction recovers the geometrical parameters of the roof and walls of a located building (Weidner, 1995). The non-buildings misclassified as buildings at previous detection stage may be found when dissimilating more information at this stage. Many algorithms use extra data such as ground plans to improve the reliability of building boundaries (Haala and Anders, 1997; Lemmens et al., 1997; Haala et al., 1998; Haala and Brenner, 1999; Brenner, 2000; Vosselman and Dijkman, 2001).
  - The model-driven algorithms are limited to process simple and specific roofs assuming flat, symmetric sloped, parametric shapes. The shape and position parameters are determined by fitting models to lidar point clouds (Weidner and Forstner, 1995; Haala et al., 1998; Maas and Vosselman, 1999). These algorithms will lead to a reliable reconstruction if all the constraints in building models are well satisfied and produce pretty results. For instance, an algorithm may assume that there exists a main orientation of the building and all edges are either parallel or perpendicular to that orientation. Actually more unrealistic constraints have been used (Vosselman, 1999; Elaksher and Bethel, 2002). But modern buildings often have structures much more complex than those models.
  - The data-driven algorithms do not assume specific building shapes for a scene, only making a few realistic assumptions such as the vertical wall constraint. A building can be expressed by bounding surfaces that may be described as planar surfaces or triangles. Although some algorithms can produce good building shapes, they often use complex plane detection techniques such as clustering of triangles based on TINs, 3-D Hough transformation and clustering of 3-D points, which often result in a heavy computational burden (Maas and Vosselman, 1999; Wang and Schenk, 2000; Vosselman and Dijkman, 2001). Some algorithms produced polyhedral models with low quality in shape (Gamba and Houshmand,

2002). My algorithm will approximate building roofs using less roof points and utilize the vertical wall constraint to polish such polyhedral models.

This chapter presents an approach for automated extraction of buildings from large lidar datasets. The basic idea is to use the geometric and radiometric information contained in lidar data to make hypothesis for the presence of buildings and to verify this using 2-D and 3-D evidences. The building extraction begins with the segmentation on the derived DNM. By developing powerful algorithms, building boundaries are reconstructed as rectangles, quadrangles or regular polygons. The buildings with flat roofs are modeled as prismatic shapes, and buildings with complex roofs are modeled as polyhedrons.

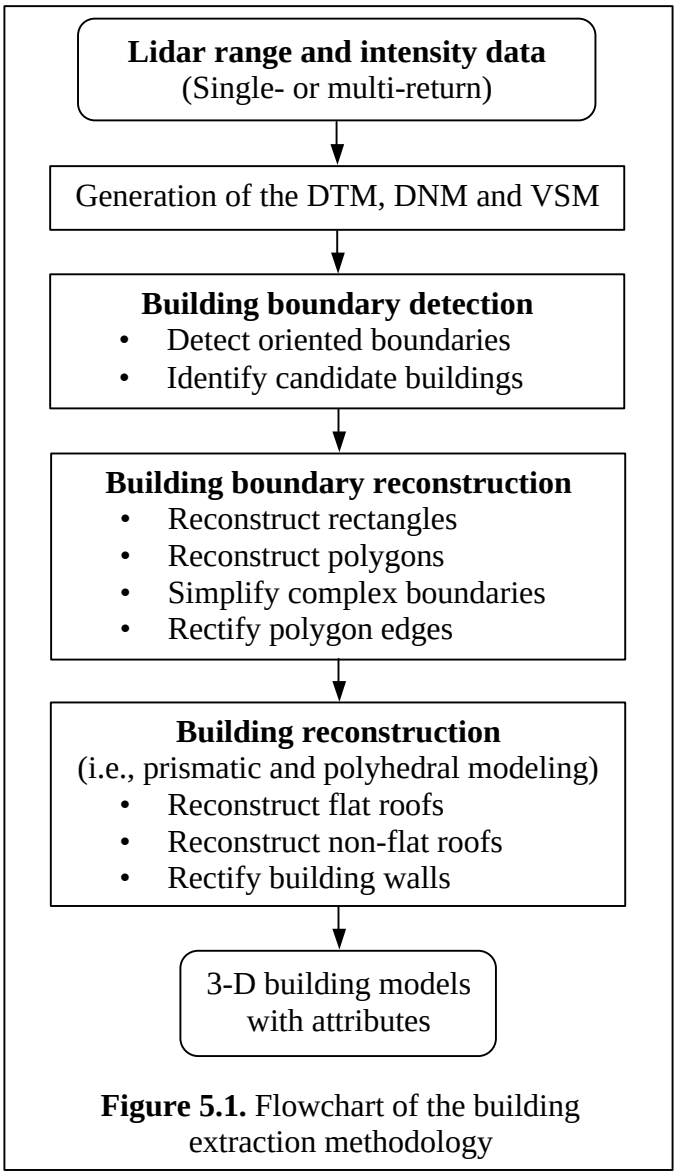


Figure 5.1. Flowchart of the building extraction methodology

5.2 METHODOLOGY OVERVIEW

Figure 5.1 shows the workflow of my methodology for building extraction. Building detection and reconstruction are the two core steps, and building detection is composed of boundary detection and reconstruction.

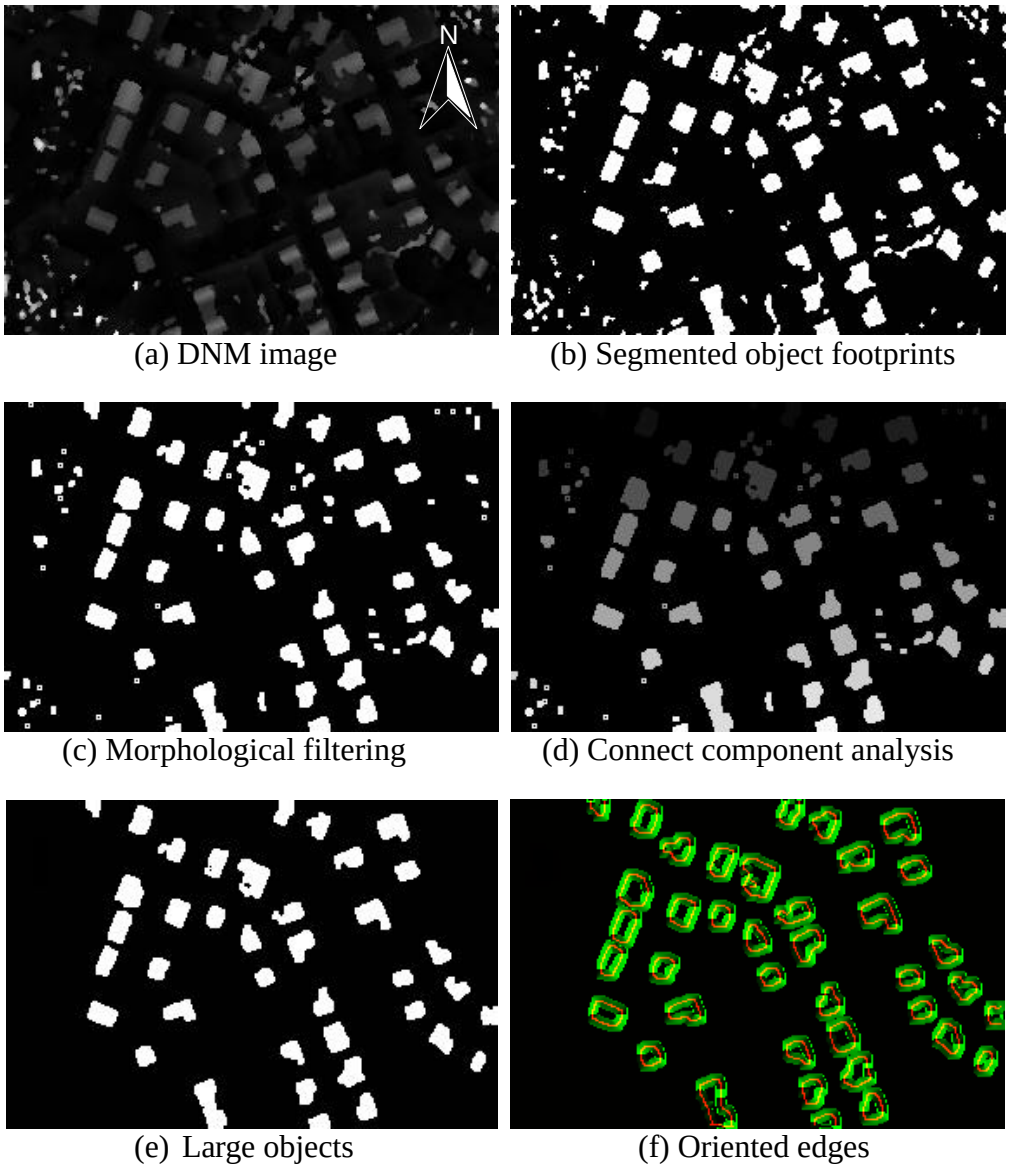
First, the DNM and VSM are generated from lidar data (see Chapter 3). The DTM is derived from last-return range data and intensity data using the HTRA, and the DNM is obtained. The VSM is derived from first- and last-return range data.

Second, object footprints are obtained by segmenting the DNM and their boundaries are detected. A boundary is represented by a chain of oriented edge pixels. Then candidate buildings are determined by a classification using shape measures.

Third, the boundaries of candidate buildings are reconstructed as polygons. A constrained searching technique is employed to find out if a rectangle represents a boundary. If not true, a sequential linking technique (SLT) is used to reconstruct the boundary as a regular polygon using an enhanced Hough transformation (EHT). These algorithms follow the hypothesis verification using 2-D cues and an optional refinement. The complex boundaries that can be reconstructed by the STL are simplified. Geometric regularity constrains are then applied to re-shape these polygons. Iterations may be needed with adaptively changed parameter values.

Finally, the flat and non-flat roof hypotheses are tested. A prismatic model is created for a flat roof building with its average roof elevation computed. Employing mathematical algorithms, numerous attributes for each building model, including area, perimeter, footprint type, footprint elevation, average or maximum roof elevation, building orientation, MBR, etc., are calculated. These candidate buildings are further verified by examining criteria of 3-D cues such as the existents of vertical walls. Non-flat roof buildings are modeled as polyhedrons by a selective refining procedure and by the rectification of building walls.

The quality of the extracted buildings is subject to some parameters used in the algorithms. The optimal values of these parameters are difficult to be determined in advance, and vary with the terrain type, coverage type and building geometry. These parameters are determined in a flexible and adaptive manner throughout the processing procedure. Their basic values are obtained by using a priori knowledge about the geometric and radiometric properties of buildings and by analyzing running results. The technical aspects are discussed further in subsequent sections.



(a) DNM image (b) Segmented object footprints

(c) Morphological filtering (d) Connect component analysis

(e) Large objects (f) Oriented edges

Figure 5.2. Object boundary detection

5.3 BUILDING BOUNDARY DETECTION

Buildings are detached objects rising vertically on all sides above the bare Earth. In geometry, building footprints are bounded by distinct edges of regular shapes relative to natural objects. Most building surfaces can be approximated by simple geometric shapes (i.e., triangles and rectangles). In contrast, vegetation clusters have extremely irregular structures and shapes. Therefore, buildings may be distinguished from vegetation objects by evaluating shape measures for their footprints (Wang and Schenk, 2000) or the roughness of point clusters (Elberink and Maas, 2000).

5.3.1 Detection of Object Boundaries

The detection of object boundaries is divided into four steps:

- *Object footprint location*. The DNM (see Figure 5.2a) is segmented into clusters that describe separate footprints of buildings and trees. The height threshold is set based on a priori knowledge (e.g., 5 m) or is calculated by an iterative selection method. The 1-valued pixels indicate the existence of objects (see Figure 5.2b). Morphological operations are applied to smooth boundaries and break the weakly connected clusters into pieces (see Figure 5.2c). Most trees may be deleted by removing those components that have relatively large intersections with the VSM (see Figure 3.8). If so, only building footprints are retained, and the shape measures based classification (see Section 5.3.2) is not necessary.
- *Small object removal*. The binary footprints are labeled by an incremental connected component analysis. Small footprints less than the a priori minimum building area are deleted to remove those small stand objects such as individual trees and trucks. Figure 5.2d shows the objects shaded by labeling values, and Figure 5.2e show the objects whose areas are larger than 70 m^2.
- *Oriented boundary detection*. The footprint boundary can be easily traced but with frequently changing directions. Here, I apply a synthesized edge detection algorithm (see Section B.1.9) to detect oriented boundaries, which are composing of oriented edge pixels as shown in Figure 5.2f. An oriented edge pixel is denoted

as $p = (x, y, b)$, which has both position (red layer) and gradient direction (green layer). The direction is calculated using Equation B.5b, and the angle is larger for a lighter gray value and is in the interval of [0^o, 180^o).

- *Oriented boundary encoding*. Each boundary is encoded in 8-connected neighborhood by chain coding (see Section B.1.10). An oriented boundary is expressed by a closed chain of orientated pixels, i.e., $B_o = \{p_i(x, y, b) \mid i = 1,..., n_B\}$, where n_B is the number of edge pixels.

An outer chain and possibly several inner parts describe a compound footprint boundary. A polygon falls inside another polygon if all of its vertices fall inside that one. In this chapter, I treat inner chains as separate boundaries.

5.3.2 Classification of Building and Vegetation Objects

Two shape measures, i.e., reducibility and circularity, are used for fast classification focusing on discarding non-building objects (mainly trees).

- *Reducibility measure* is the ratio of the number of points n_G in the generalized chain and the number of points n_B in the original chain as given by Equation 5.1a. A smaller value indicates a higher possibility of belonging to the building class.

$$SM_{reducibility}(B) = \frac{n_G}{n_B} \tag{5.1a}$$

- *Circularity measure* is the ratio of the squared perimeter of the chain over its area as given by Equation 5.1b (Parker, 1993; Wang and Schenk, 2000). For a circle, the ratio is $4p$, and for a square, it is 16. A larger value indicates a higher possibility of belonging to the building class.

$$SM_{circularity}(B) = \frac{Perimeter^2(B)}{Area(B)} \tag{5.1b}$$

The membership function of a boundary to be classified to the building class is obtained by combining the above two shape measures as given by

$$BC_{SM}(B)=\begin{cases}1 & SM_r \le Tr \wedge SM_c > Tc \\ 0 & SM_r > Tr \wedge SM_c \le Tc \\ [1-(SM_r-TR)]\times(SM_c-4p)/(16-4p) & otherwise\end{cases} \quad (5.2)$$

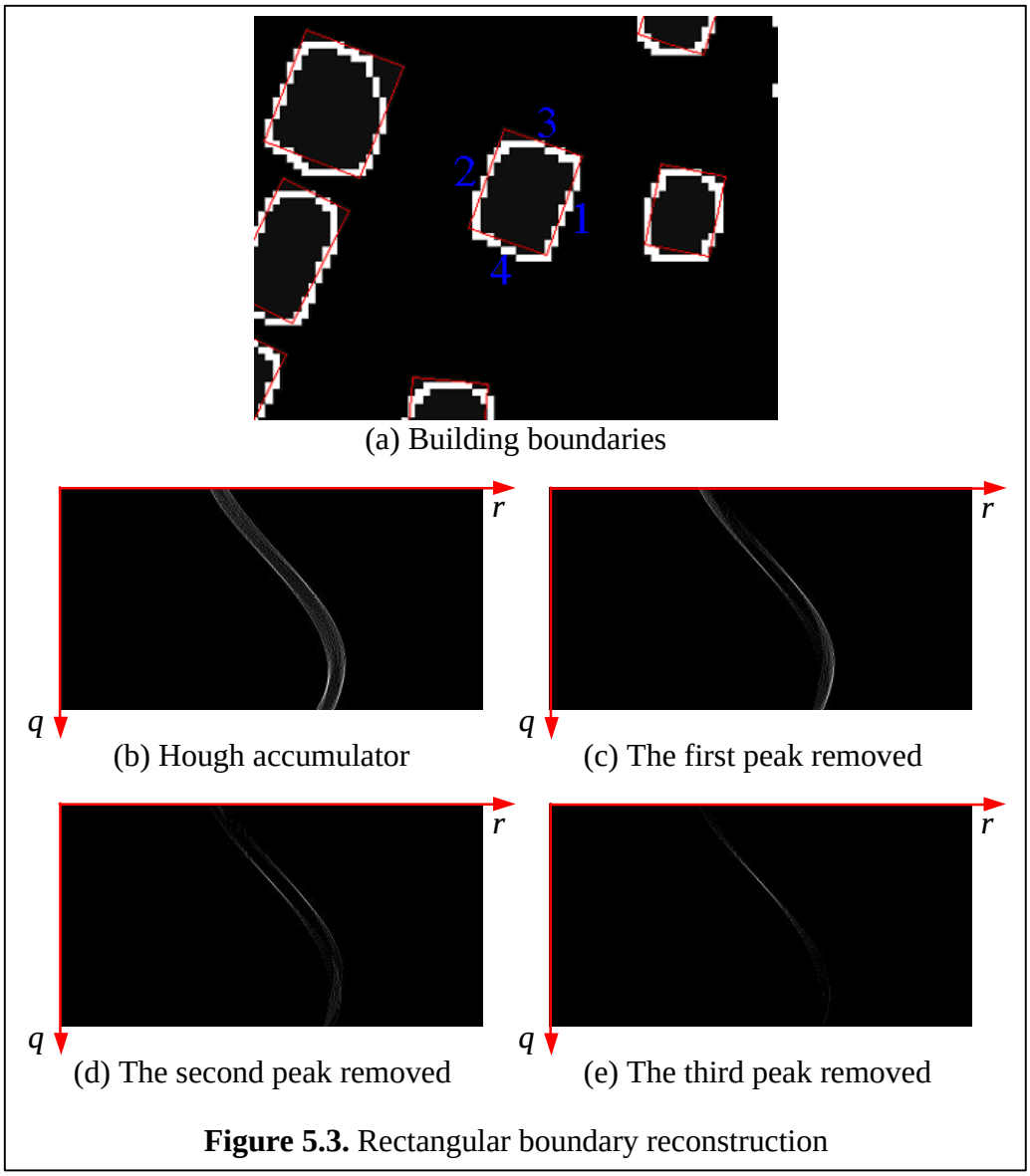
where *Tr* and *Tc* are thresholds that are computed by iterative selecting method over the reducibility and circularity measures, respectively, of all the boundaries. Except the reducibility and circularity measures, other shape measures such as symmetry could be used to classify specific shape types. If a building is occluded partly by trees, it is may be wrongly classified to the tree class as shown in see Figure 5.5a (the green polygon).

5.4 RECTANGULAR BOUNDARY RECONSTRUCTION

A rectangle consists of two pairs of parallel lines, and one pair of parallel lines must be perpendicular to the other pair. When the edge points of the boundary are processed using Hough transformation, this geometric constraint appears as two pairs of peaks located inside two slim zones along two angles in Hough space, and the angle difference is 90^o. So I design a constrained searching procedure to detect a rectangle reliably using the SHT (see Section 4.4.1). A part of edge map in Figure 5.2f is magnified in Figure 5.3a, where the white pixels represent the footprint boundary. The Hough transformation result of the boundary in the center is shown in Figure 5.3b.

First, four lines $(r_i, q_i), i = 1,...,4$ are detected in both Hough space and image space using the rectangle constraint and form the rectangle hypothesis. The first line corresponds to the largest peak, which is detected as shown in Figure 5.3c. The second line is parallel to the first line, and its corresponding peak must lie inside a slim buffer along the first line's angle, and is deleted in Figure 5.3d. The third line and the fourth line are similarly searched in a slim buffer along an angle that is shifted 90^o from the mean angle of previous two lines as shown in Figure 5.3e, where the third peak is deleted. Occasionally,

two lines that are parallel to the *y*-axis may present a spurious and inconsistent condition. That is, one line's angle is close to 0^o, while another is close to 180^o. In this situation, both angles are corrected to a mean angle that is shifted 90^o from other two lines. The four corners are obtained by intersecting those four Hough lines using Equation 4.5.



(a) Building boundaries

(b) Hough accumulator

(c) The first peak removed

(d) The second peak removed

(e) The third peak removed

Figure 5.3. Rectangular boundary reconstruction

Next, the hypothesized rectangle model is verified. A complex check regarding several evidences is performed to judge if the four lines form a rectangle of good quality.

- *Area evidence* is measured by the ratio between the area values computed for the hypothesized rectangle and the raw boundary chain, which should be close to 1.

- *Perimeter evidence* is measured by the ratio between the two perimeter values computed for the hypothesized rectangle and the original boundary chain, which should be close to 1.
- *Dimension evidence* is the measurement of length differences, which should be less than certain thresholds. There are three types of quantities available for the length (or width): the absolute difference between the distance parameters of the third and the forth lines (or of the first and the second lines), the average peak values in Hough space for the first and the second peaks (or for the third and the fourth peaks), and the average number of grouped points for the first and the second lines (or for the third and the fourth lines) because each line contains a sequence of edge points connected in 8-neighborhoods.

Finally, a validated rectangle model is rectified by adjusting the parameters of its four lines to make them form a perfect rectangle with four internal right angles. The weighting factors in least-squares adjustment are determined according to length and width attributes. Several rectangular boundaries are shown in Figure 5.3a, where the red rectangles are reconstruction results.

A rectangle reconstructed this way does not need any additional rectification (see Section 5.7). The rectangle model may be created in alternative approaches. For example, I can detect the first largest four peaks to form the hypothesized rectangle; then the parallelism constraint and the above evidences are verified. In addition, a footprint with four lines can also be reconstructed using the SLT described in Section 5.6.

5.5 ENHANCED HOUGH TRANSFORMATION

The 2-D HT transforms each pixel from the image domain into a curve in Hough space (see Section B.1.12). Usually there are many types of buildings whose boundaries can be divided into many pieces of line segments. So the Hough accumulator may present many peaks. The detection of these peaks is equivalent to the division of the boundary points into different groups of line points. Since an oriented boundary is composed of a chain of

successively connected oriented edge points (see Figure 5.4a), the SHT can be enhanced to reduce the possibility of wrong grouping by using the direction information. In this section, I will propose an enhanced Hough transformation, which is composed of three processing steps, i.e., grouping, merging and splitting. After analysing a boundary using the EHT, a building boundary will be divided into a set of edge segments each attached with a group of edge points.

5.5.1 Grouping Line Points

With the peak parameters (r,q), the buffer of points belonging to a line is determined by $(r-s_r, q \pm s_q)$ and $(r+s_r, q \pm s_q)$, where s_r and s_q are distance and angle tolerances (Zhang and Burkhardt, 2000). In my experiments, the buffer is limited to be a slim zone around the detected Hough line with the angle tolerance omitted. But another tolerance regarding the direction constraint is introduced. In a word, the line points should fall inside this buffer and have close gradient direction. In addition, to reduce the computational burden, only a sub-range of $[0^o, 180^o)$ around a point's direction is accumulated. Let $p(x, y, b)$ denote an oriented point of the raw boundary *B*, and $l(r,q)$ a line. The distance between *p* and l is given by

$$d(p,l) = | x\cos q + y\sin q - r | \qquad (5.3)$$

The following criteria are used to search associated points in the boundary:

- If $d(p,l)$ is smaller than a small threshold (e.g., 1 pixel), the difference between the point's gradient direction and the line's angle should not be close to 90^o, i.e., $| b - q | < 90^o - s_b$ or $| b - q | > 90^o + s_b$, where s_b is a small tolerance (e.g., 15^o).
- If $d(p,l)$ is larger than the small threshold but smaller than a larger threshold (e.g., 2 pixels), the point's direction should be close to the line's angle, i.e., $| b - q | < 90^o - s_b'$ or $| b - q | > 90^o + s_b'$, where s_b' is a large tolerance (e.g., 45^o). This will find distorted pixels in edge detection for non-idea straight lines.

By introducing direction information, an edge point would not be assigned to a Hough line if the angle deviation were large. As shown in Figure 5.4b, the Hough line pointed by the arrow will include the points enclosed in both the two ellipses and the two circles by the conventional searching method. While the points in the two circles actually belong to s_5 and s_6, respectively. My method will group only the points in the ellipses according the above criteria since the direction of the points in the circles (i.e., 145^o) is significantly different from the angle of that Hough line (55^o). This precise grouping is very beneficial for obtaining correct linking order in subsequent permutation of edge segments.

The grouping procedure is continued until the termination condition is met. Multiple criteria combining several conditions are used to check if this sequence of Hough lines approximates the original boundary well. For example, the remaining points in the chain are below a certain percentage, the line detected recently is longer than a predefined threshold and comparable with previously detected lines. These thresholds are determined according to the number of lines and the footprint type.

For a long line, its parameters can be refined by fitting grouped points to a line equation as given in Equation 5.4a or 5.4b depending on to its angle. The line parameters searched in Hough space are replace with the fitted parameters if they are very close, for example, the angle and distance differences are smaller than 2.5^o and 5 m, respectively. In addition, line points can be re-grouped in image space using the updated line parameters.

$$a_1 x + y + b_1 = 0 \ (a_1 \le 1), \quad 45^o \le q < 135^o \tag{5.4a}$$

$$x + a_2 y + b_2 = 0 \ (a_2 \le 1), \quad q < 45^o \vee q \ge 135^o \tag{5.4b}$$

5.5.2 Merging Lines

An over-division of the boundary may happen when the grouping procedure detects more lines than those actually exist in the boundary. To tackle this problem, each line detected recently is compared with all the lines detected before. Two lines that are close will be merged to eliminate the confusing, and their edge points are also merged. The angle parameter q of the new line is determined by a LSs adjustment weighted by the lengths

of its two components; then the distance parameter r is re-calculated by averaging the distance values computed using Equation 5.2 for all the associated points. The following two criteria determine two close or collinear lines:

- *Angle evidence* is measured by the difference between their angle parameters.
- *Distance evidence* is measured by the difference between their distance parameters.

The difference for each parameter should be smaller than a given tolerance, that is, 7.5^o for s_q and 5 m for s_r. To ensure that the edge points in a merged segment are still sequential, they are re-permuted so that their coordinates change monotonously.

5.5.3 Splitting Edge Segments

It is easily observable that the points of a Hough line may contain multiple separate edge segments that are collinear as shown in Figure 5.4b, where the Hough line pointed by an arrow is composed of two edge segments s_7 and s_8 that have a same pair of line parameters. Because the points are grouped following the coding order, the edge points associated with a line must follow the same successive order of the original chain. It is reasonable to require that these successive points belong to a separate edge segment should be all connected in 8-neighbourhood or only have a small distance (e.g., 5 m) between them, or they belong to two different segments. Therefore, a line can be divided into multiple shorter segments by carefully checking the internal structures of those points. An edge segment s is represented by its parameters (r,q), and the associated edge points (x_u, y_u, b_u), $u = 1,...,m$, where m is the number of edge points of that segment. Those segments having only a few (e.g., 3 or less) points are discarded.

5.6 POLYGONAL BOUNDARY RECONSTRUCTION

The non-rectangular building boundaries can be represented as simple polygons using a sequential linking technique as shown in Figure 5.4. This linking procedure employs the EHT to detect edge segments, and works as follows:

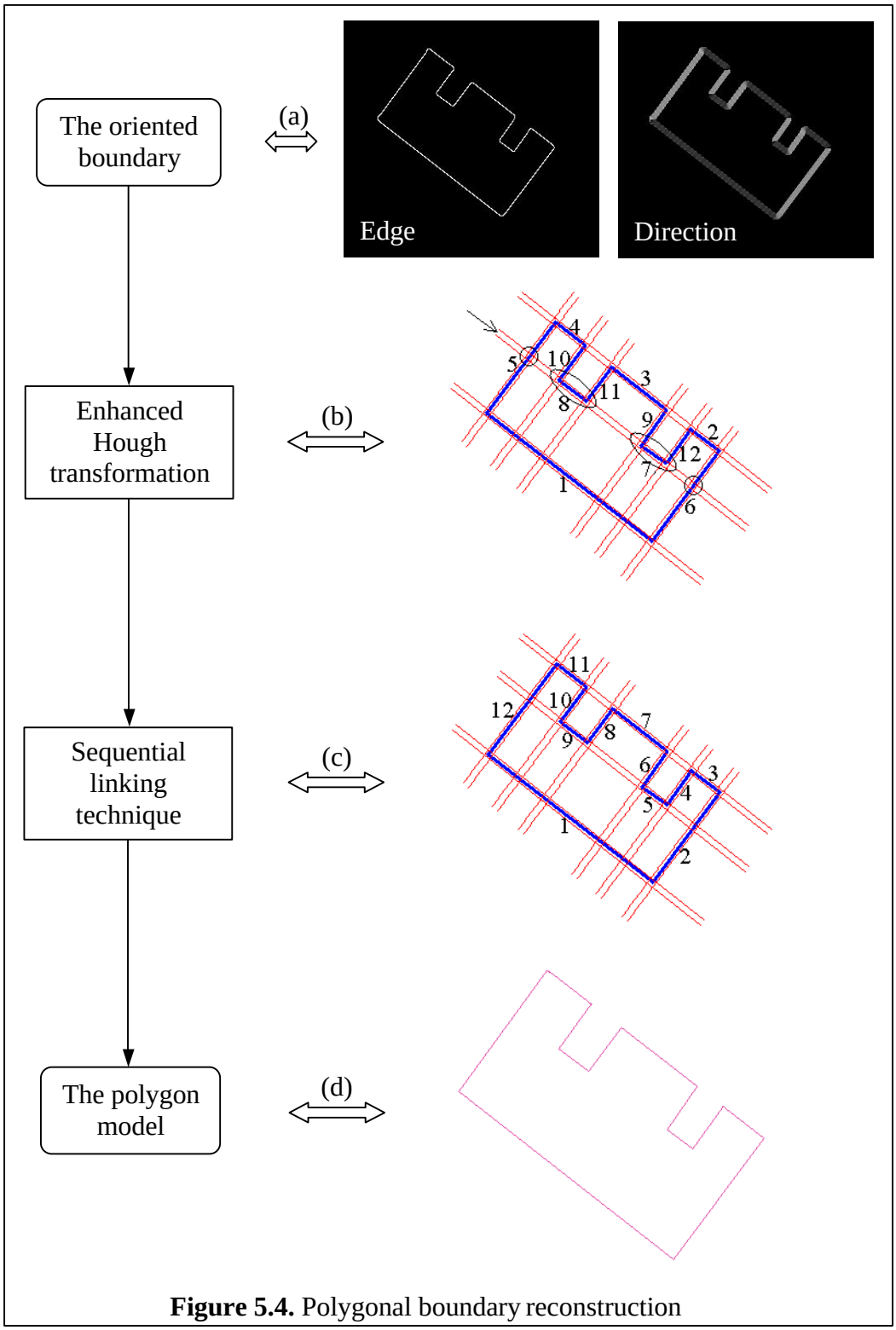


Figure 5.4. Polygonal boundary reconstruction

1. Detect all the edge segments for an oriented boundary using the EHT, i.e., $B_o = \{s_i, i = 1,...,n\}$, where n is the number of segments in the boundary. For

example, the boundary shown in Figure 5.4 has twelve edge segments. The distance between two segments s_i and s_j is defined as the smallest distance among the two point sets associated to them, and is given by Equation 5.5, where $d(p_u, p_v)$ is the Euclidian distance between two points p_u and p_v.

$$d(s_i, s_j) = \min_{p_u \in s_i \wedge p_v \in s_j} d(p_u, p_v) \tag{5.5}$$

2. Form a complete boundary by linking the edge segments. Some quantities are recorded during the linking procedure, including the minimum and second minimum distances and angle differences between adjacent segments (including the last and first segments). An empty linking list is initialised at the beginning.
   2.1. Find the first reference segment. It is the longest segment, and is placed into the head of the linking list. The orientation of this segment is often the main orientation of a building, and is 139^o for s_1 in Figure 5.4b.
   2.2. Search the next reference segment based on the current one until all the segments are linked one by one.
      2.2.1. Remove the current reference segment from the set of segments in B_o.
      2.2.2. Calculate the distances between the reference segment and the remaining segments in B_o using Equation 5.5. Following criteria are applied to select the next reference segment.
         - Early skipping. If the minimum distance is larger than a small threshold (e.g., 3 m) and the second minimum distance is larger than a large threshold (e.g., 5 m), then exit the linking procedure.
         - If the minimum distance is unique, then the corresponding segment is selected.
         - If several distances are equally small or less than the small threshold, then the segment most perpendicular to the reference one is selected.
      2.2.3. Append the selected segment into the linking list.

   At this point, a boundary is expressed as a sequence of edge segments, which are

numbered in correct order, for example, as shown in Figure 5.4c.

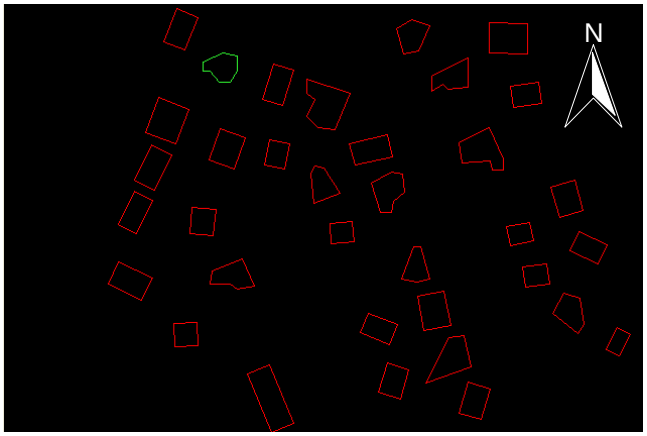
3. Form the polygon hypothesis (see Figure 5.4d). The polygon vertices are obtained by intersecting all pairs of adjacent segments in the linking list using Equation 4.5. The first and the last segments are also intersected. Two adjacent vertices whose distance is smaller than a threshold (e.g., 3 m) are merged.
4. Verify the hypothesized polygon. A qualified polygon satisfies two conditions:
   - *Simplicity*. The polygon is not valid if it is self-intersected or there are intersected vertices outside of a buffer of the MBR of that footprint.
   - *Integrity*. Besides the area and perimeter evidences (see Section 5.4), a few more criteria are used:
     * *Angle evidence* is measured by the minimum internal angle of the hypothesized polygon, which should be larger than a predefined angle threshold (e.g., 15^o) for a qualified polygon.
     * *Linkage evidence* is given by the maximum and the secondary maximum distance between all the pairs of adjacent segments, which should be is less than a threshold (e.g., 5 m).
     * *Distance evidence* is measured by the Hausdorff distance (Rucklidge, 1997; Jian et al., 1998) between the hypothesized polygon and the original boundary, which should be less than a threshold (e.g., 8 m).

Figures 5.5b to 5.5e show some linked polygons. For a quadrangle, this linking procedure is not necessary. An alternative way to determine a proper linking order of its four segments is to find out the two most parallel segments first and then re-permute the other two segments.

5.7 POLYGONAL BOUNDARY RECTIFICATION

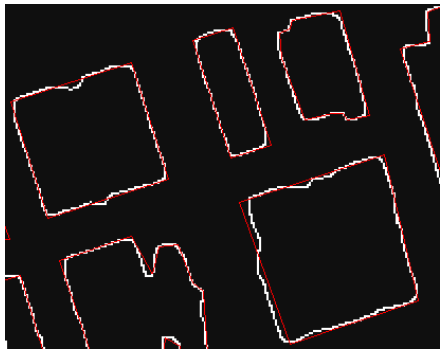
Complex building boundaries that cannot be reconstructed by the constrained searching or the SLT are still represented by pixel chains with very low cartographic quality. I generalize these boundaries by enforcing some distance and angle constraints using the improved Douglas-Peucker simplification based on convex hull (see Section B.2.3) and

straight-line detection techniques (Zhang, 1993). The termination condition for the simplification is that the area of the generalized polygon be close to that of the boundary.

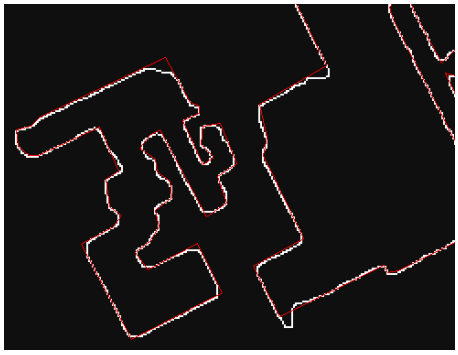


(a) Village site

(b) U-structures



(c) L-structures



(d) Mixed U- and L-structures

(e) Mis-linkage

Figure 5.5. Building boundary models

Before this point, no specific a priori knowledge about the geometric regularity characteristics (e.g., parallelism, orthogonality and collinearity) of typical building shapes is systematically applied to individual structures of a polygon model. The polygons may

have arbitrary angles and segment lengths. In man-made environments, however, long straight segments and right angles are often the realism. I try to rectify those polygons with five or more vertices by imposing multiple angle and length constraints, still allowing for non-orthogonal, non-straight angles and short segments. The following rules try to recover aforementioned geometric regularity characteristics and are iteratively applied to all the structures of a polygon.

- *U-structures* are every three consecutive segments in a polygon. Both parameters (r,q) of the segments will be updated during the adjustment, and the related vertices are accordingly replaced by new intersections of updated lines.
  - *Parallelism*, i.e., the angle difference between first and the third segments is smaller than a threshold (e.g., 3^o).
    * *Rectangular corners*. If both internal angles are nearly 90^o, then the angles of these segments are adjusted according to their lengths so that the first and the third segments becomes parallel and the second one is perpendicular to them.
    * *Non-rectangular corners*. If neither angle is 90^o, then the first and the third segments are adjusted to be parallel weighted by their lengths.
  - *Corner distortion*. This assumes that the second segment is shorter than a predefined length threshold (e.g., 5 m) or a rate of the longest segment.
    * *Orthogonality*. That is, the angle between the first and the third segments is nearly 90^o. The first and the third segments are adjusted to become exactly orthogonal and the second is dropped. This case corrects distorted round corners due to the smoothing effect in edge detection.
    * *Collinearity*. If the angle between the first and the third segments is far away from a right angle and the angle between the second and either longer segment is approximately 180^o, then the short one is dropped and the longer one is extended with its orientation slightly adjusted.

- *L-structures* are every two consecutive segments in the polygon.
  - *Orthogonality*. If the angle is nearly right, then adjust the two segments to make them orthogonal and intersect a new vertex to replace the old one.

- *Collinearity*. If the internal angle is approximately 180^o, then drop the intermediate vertex.

• *Quadrangles*. If a rectified polygon has four vertices, then try the rectangle reconstruction for the corresponding boundary with larger thresholds. A few rectangles missed before may be detected.

Figure 5.5 shows some rectified cases. The rectified polygons in Figure 5.5a correspond to the boundaries in Figure 5.2. The rectification cannot correct errors resulted in the sequential linking as shown in Figure 5.5e, where occasionally, the reconstructed polygons may be mis-linked in wrong order.

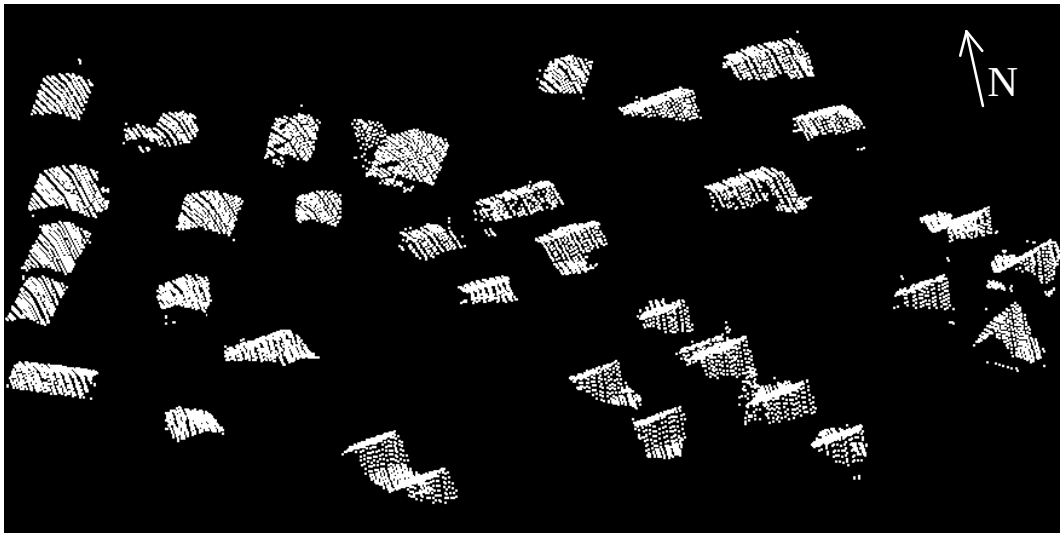
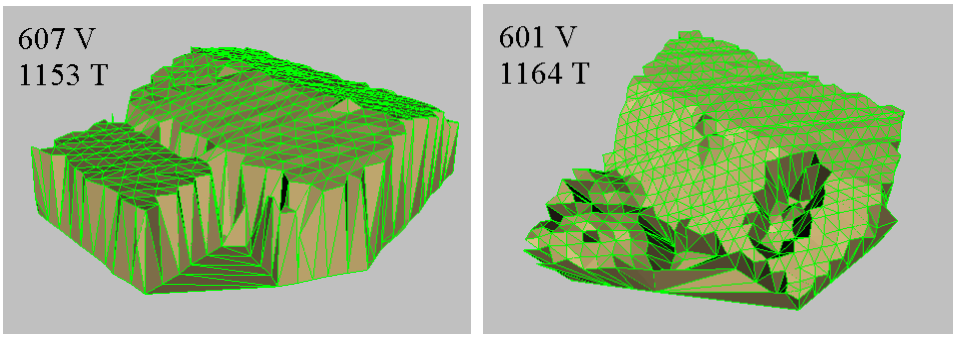


Figure 5.6. 3-D view of roof points (Village site)

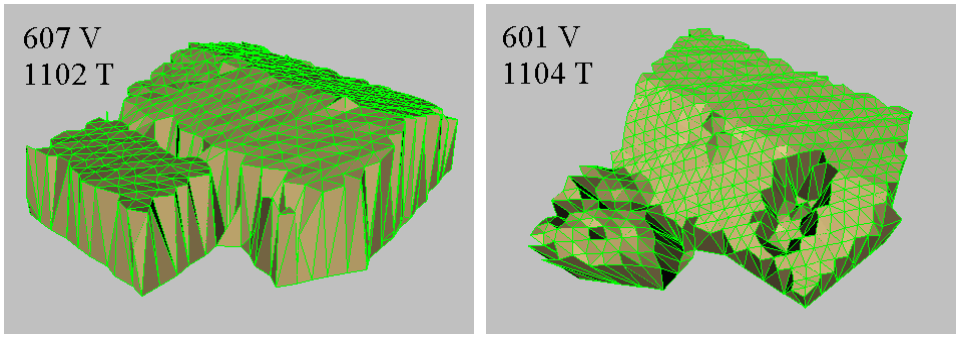
5.8 BUILDING RECONSTRUCTION

The lidar points hit on a building's roof can be retrieved from the raw lidar points by doing point-in-polygon tests (see Section B.2.2) using its polygon model. Figure 5.6 gives a 3-D view of roof points inside the polygons shown in Figure 5.5a. At this point, a building model consists of a polygonal boundary and a number of roof points, which can be triangulated to give a rough delineation of the whole outer surface of a building including its footprint, roof and walls. TIN is widely used to model surfaces (see Section

B.2.4). Two buildings are triangulated in 2-D and are then visualized in 3-D as shown in Figure 5.7a. To represent the building's surface, I remove redundant edges, a part of which are outside the polygon in 2-D. Figure 5.7b shows the corresponding constrained TIN representations. However, this kind of approximation still exhibits too many vertices and facets, and their direct use in GIS will lead to inefficient manipulations.



(a) TINs of building surface points



(b) Constrained TINs

Figure 5.7. Triangulated building surface models

5.8.1 Roof Hypotheses Tests

I classify building roofs into two classes, i.e., flat and non-flat, which correspond to two generic shapes, i.e., prismatic and polyhedral models. The prismatic model represents a building as one flat roof polygon and one footprint polygon with a set of vertical walls. The polyhedral model approximates a building using bounding surfaces together with their intersections and topology relations.

The flat hypothesis is firstly tested for each building. I perform the a-trim operation over all the roof points to remove spurious points not returned from the roof, including those

hit on walls, on temporary structures erected over the roof and occasional on the ground. The roof points are assumed to have a homogeneous 2-D distribution, and their elevations are assumed to belong to a normal distribution for a flat roof, that is, $x \sim N(a, s^2)$, where a and s^2 are the unknown mean and variance. The test statistic T_{c^2}, the null hypothesis H_0, and the alternative hypothesis H_1 are given by (Migon and Garmerman, 1999)

$$T_{c^2} = \frac{1}{s_0^2}\sum_{i=1}^{m}(x_i - \bar{x})^2 = \frac{(m-1)s^2}{s_0^2} \sim c^2(m-1) \tag{5.6a}$$

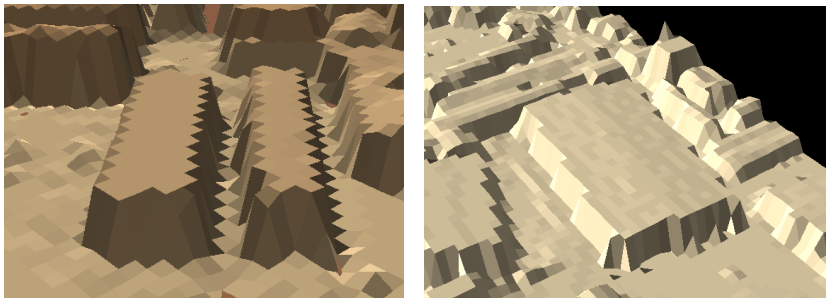
$$H_0 : s^2 \le s_0^2, \qquad H_1 : s^2 > s_0^2 \tag{5.6b}$$

where s_0^2 is the population variance factor, $\bar{x}$ is the sample mean, s^2 is the sample variance, m is the number of roof points, and c^2 denotes the chi-square distribution.

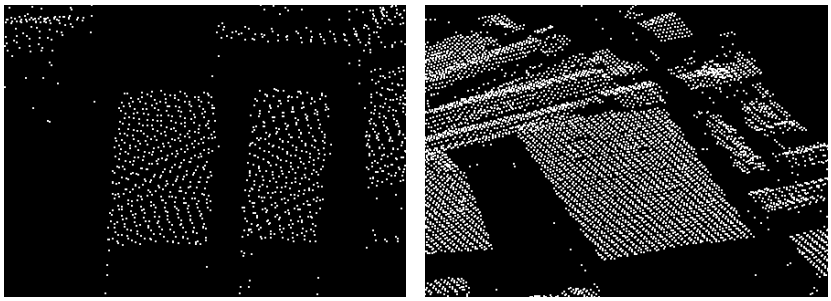
The null hypothesis H_0 claims that the roof is flat with trivial undulations. The variance factor s_0^2 can be adaptively modified by enforcing some rules. If the details of a big and tall building are not important, then I may allow for larger roof undulations. For example, it could be 1 m^2 when considering the necessary superstructures placed on the roof. Extra conditions may be combined to help discriminate if a roof is flat. For example, the sample range of the roof points should be less than a threshold that is determined by a ratio of the building's average height and the median height of all the buildings in the whole region. The flat roof hypothesis can also be tested by fit a plane equation to the roof points and check the standard deviation of the residuals. This is also suited to test sloping roofs. If the roof elevation is known to be a_0, the test statistic T_t, the null hypothesis H_0, and the alternative hypothesis H_1 are given by Equation 5.7, where t denotes the Student distribution (Zhang, 1993).

$$T_t = \frac{\bar{x} - a_0}{s/\sqrt{m-1}} \sim t(m-1) \tag{5.7a}$$

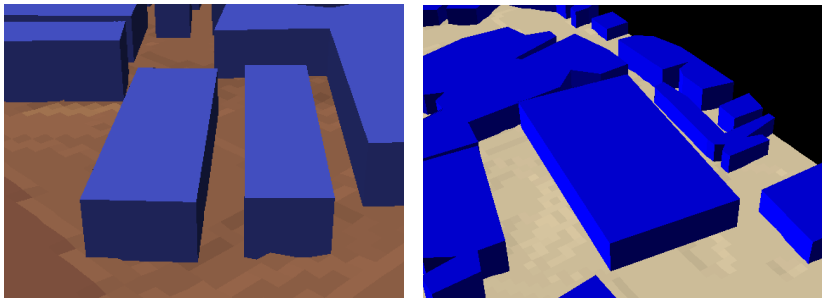
$$H_0 : a = a_0, \qquad H_1 : a \ne a_0 \tag{5.7b}$$



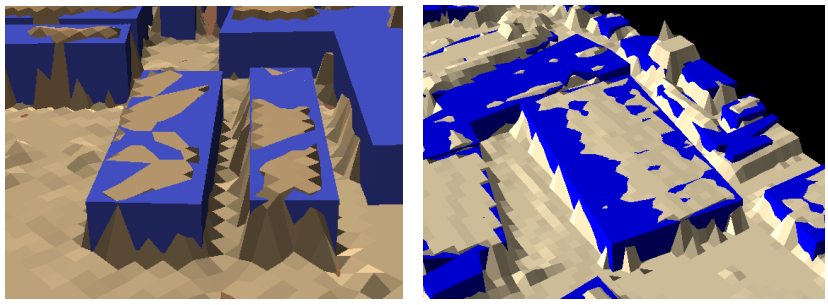
(a) Shaded lidar DSMs



(b) Building roof points

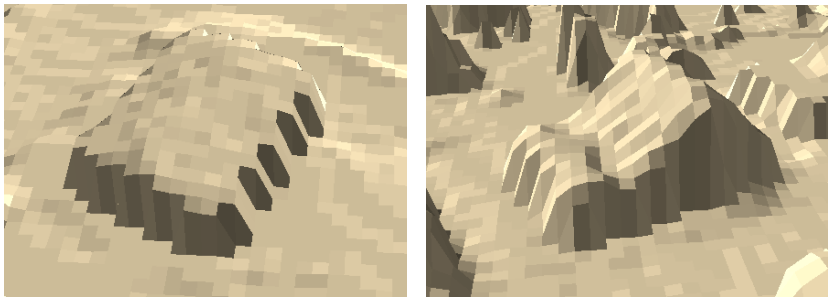


(b) Prismatic models

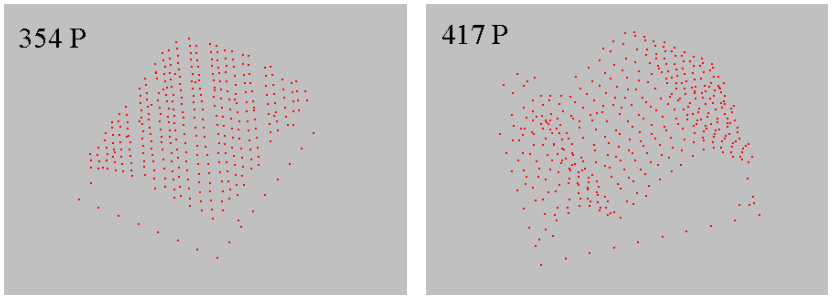


(c) Prismatic models embedded into the DSM

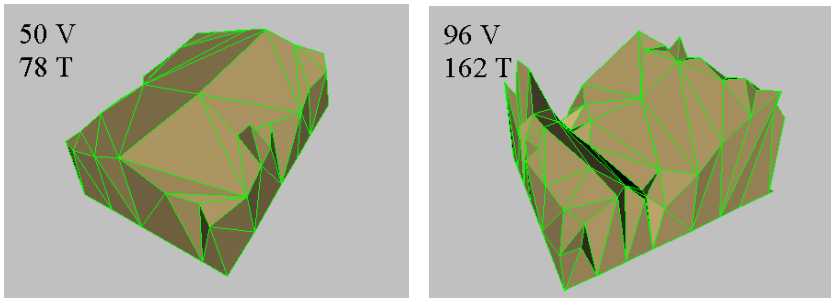
Figure 5.8. Prismatic models for flat roof buildings



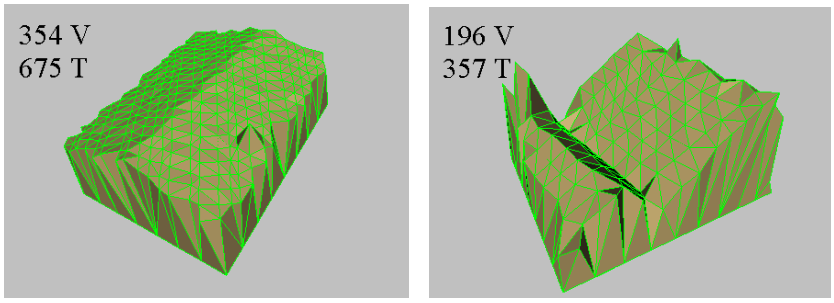
(a) Shaded lidar DSMs



(b) Roof points, polygon vertexes and interpolated vertexes



(c) Polyhedral models



(d) Finer polyhedral models

Figure 5.9. Polyhedral models for non-flat roof buildings

5.8.2 Prismatic and Polyhedral Model Generation

If the null hypothesis H_0 in Equation 5.6b is accepted at the significance level of 10%, then the building is modeled by a prismatic shape. The building roof has the same shape with the footprint with all the vertices offset to the average roof elevation. The walls are vertical trapeziums each formed by two consecutive vertices of the footprint polygon and their corresponding vertices of the roof polygon. The wall evidence is the presence of wall points that fall inside the buffers of some hypothesized wall polygons. Additional conditions may be used. For example in a single strip of lidar points, one or two adjacent walls among the four walls of a rectangular building may find wall points; while their opposite walls must have no wall points. Figure 5.8 shows the reconstruction of prismatic models for two flat roof buildings.

If the alternative hypothesis H_1 in Equation 5.6b is accepted at the significance level of 10%, then the building has a polyhedral shape. I reconstruct the roofs and walls using a selective refining approach. The reconstruction begins with a simplified model with only essential points, and iteratively refines the model by selecting the best points.

First, a rough model is created to include the highest roof point, the vertices of the footprint polygon and intermediate vertices interpolated on the polygon's edges. The number of intermediate vertices depends on the ratio between the length of an edge and the GSD. Using an incremental Delaunay triangulation algorithm, a constrained TIN is generating in 2-D. Then the edges connecting consecutive vertices on the boundary (i.e., polygon vertices and interpolated inter-mediate vertices) are inserted first, and the roof point is inserted thereafter. The details of this algorithm are described in Vigo (1997).

Next, the rough model is refined by inserting best points iteratively. For each triangle in the constrained TIN, a critical point is identified. It is a roof point whose 3-D distance from the triangle's plane is largest among all the roof points falling inside this triangle in ground plane. The critical point is inserted as a new vertex in the TIN only if the distance is larger than a threshold. The threshold may be determined adaptively according to the relative size between the building and the median size of all the buildings in the whole

region. As before, a larger threshold may be used for a big building to omit unnecessary details. Only triangles newly added are considered. This step is repeated until no more critical points can be added. In addition, one iteration is enough if the inter-mediate vertices are interpolated densely. So a polyhedral model with moderate details is generated to approximate the building surfaces. Figure 5.9 shows the reconstruction of two polyhedral building models for Village data.

Finally, the rough polyhedral models are rectified by imposing multiple constraints to recover the geometric regularity of typical buildings. My method is to adjust the coordinates of ridge vertices in the TIN according to the vertical wall constraint. A roof point is defined as the ridge vertex if there are one or more edges connecting that roof vertex and boundary vertices. In this situation, I say that the ridge vertex and the connected boundary vertices are visible. In Figure 5.10, three types of ridge vertices, including I-vertex (red dot), V-vertex (blue dot) and Y-vertex (yellow dot), are identified around one corner of a building from Village data.

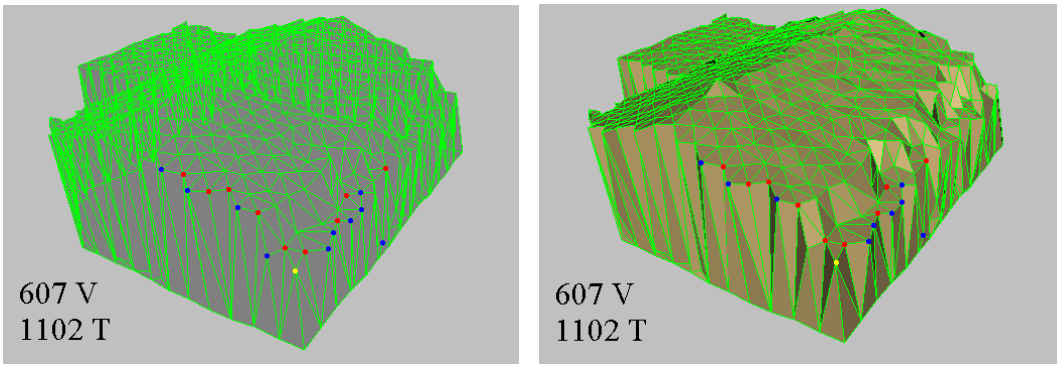
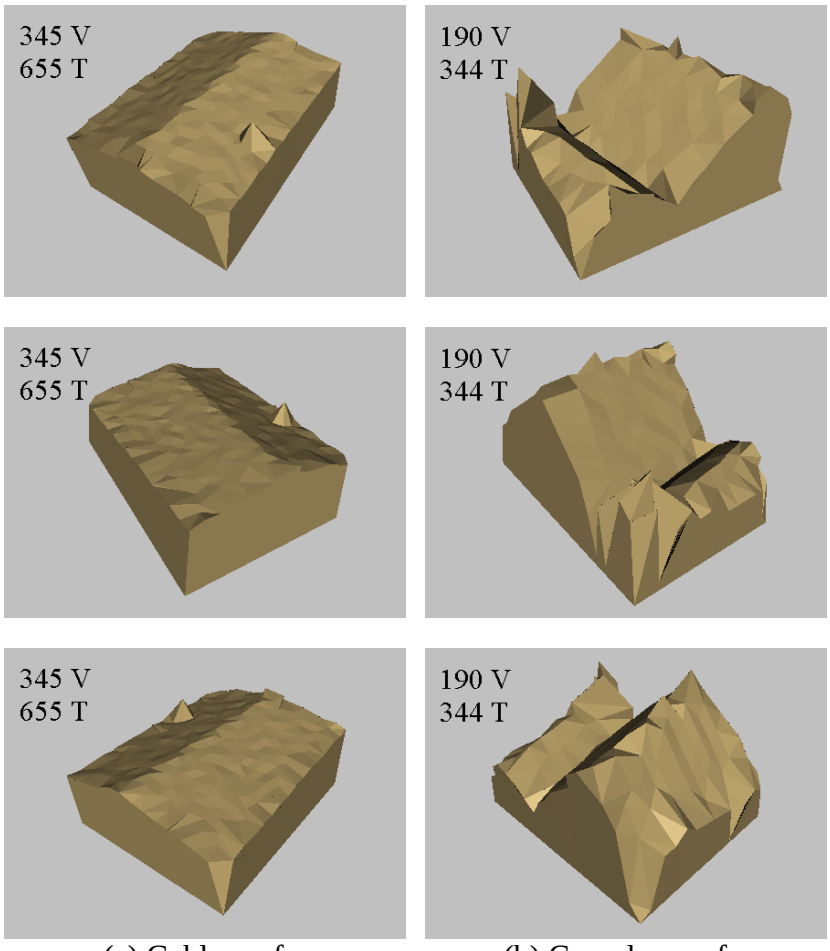


Figure 5.10. Three types of ridge vertices

- *I-vertex* is a ridge vertex visible with only one boundary vertex. Its coordinates are replaced by those of the intersection point between a vertical line passing through the associate boundary vertex and a roof triangle. The selected roof triangle has three roof vertices including the I-vertex. When a boundary vertex is visible with multiple I-vertices, only the first I-vertex is kept and others are removed. The edges associated with other I-vertices are changed to connect to the first one, and duplicate edges are also removed.

- *V- vertex* is a ridge vertex visible with two adjacent boundary vertices. Its position will move to that of an intersection point between a vertical line and a selected roof triangle. The vertical line passes through the centre of the edge connecting the two boundary vertices.
- *Y- vertex* is a ridge vertex visible with three consecutive boundary vertices. It usually occurs around right corners, and the second boundary vertex is sited at the corner. A Y-vertex will move to the intersection point between a vertical line and a selected roof triangle. The line passes through the second boundary vertex.



(a) Gable roof (b) Complex roof

Figure 5.11. Different views of rectified polyhedral models

The two buildings shown in Figure 5.9 are rectified, and their different views are shown in Figure 5.11. The rectified polyhedrons have good shape for most walls except few distortions, for example, at corners. However, these polyhedral models are sufficient for

many applications such as telecommunications.

5.9 TEST RESULTS AND EVALUATION

Three datasets collected at residual and urban areas, including Village site (see Section 3.6.1.1), Osaka city (see Section 3.6.1.2) and downtown Toronto (see Sections 3.6.1.3 and 4.2.1), are used to test the proposed building extraction algorithms.

5.9.1 Building Extraction Results

Same rules are used to obtain all the threshold parameters with fixed initial or adaptively changing values. That is, I do not set special parameter values different for one dataset from that for another.

5.9.1.1 Results of Village Site

In Figure 5.12, the prismatic building models together with the lidar DSM are shown with three-time exaggeration in vertical direction. Most buildings are correctly extracted and are coloured by their footprint types, including rectangle, quadrangle, polygon and complex. All the building roofs are classified as non-flat surfaces.

5.9.1.2 Results of Osaka City

In Figure 5.13, the prismatic buildings together with the lidar DSM at the bottom part of the scene are shown. The building models are coloured by their roof types, including 33 flat roofs and 150 non-flat roofs. Most building footprints are modeled by rectangles and regular polygons.

5.9.1.3 Results of Downtown Toronto

In Figure 5.14, the building polygons detected (green lines) and those surveyed polygons in the DLGs (red lines) are overlaid on the DOQs and the DNM, respectively. The prismatic building models overlaid upon the derived DTM are shown in Figure 5.15, where the models are coloured by their footprint types.

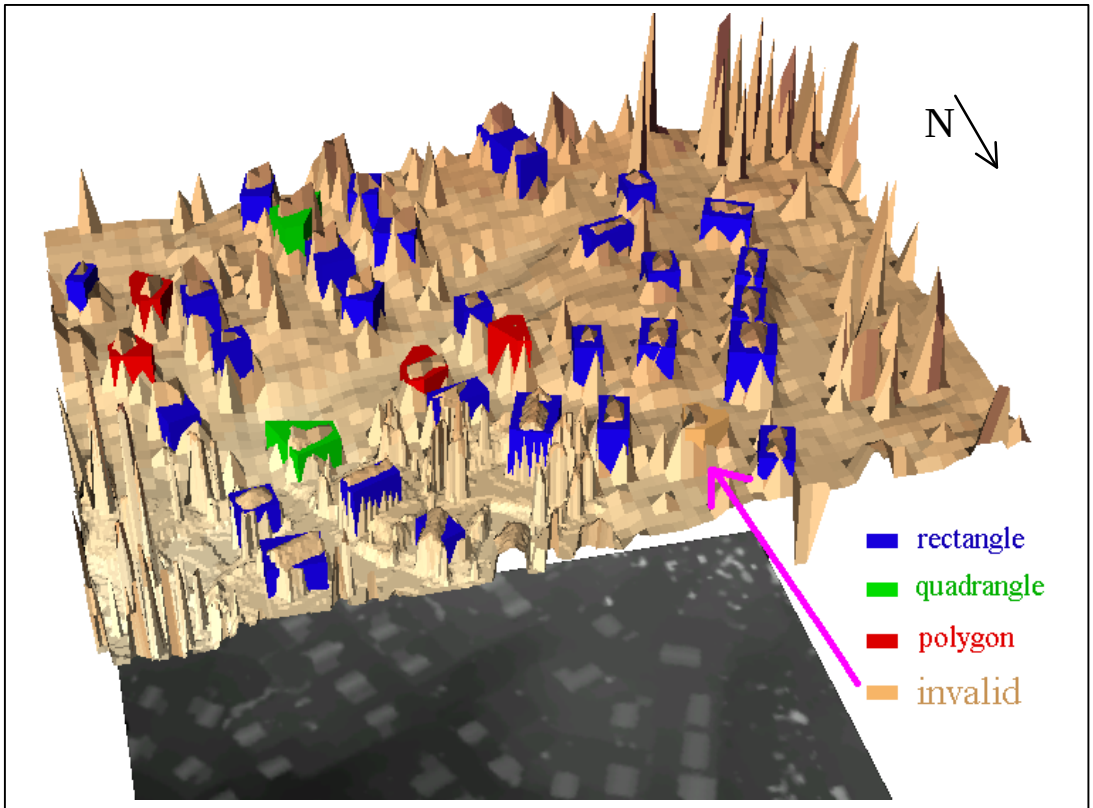


Figure 5.12. 3-D building models (Village site)

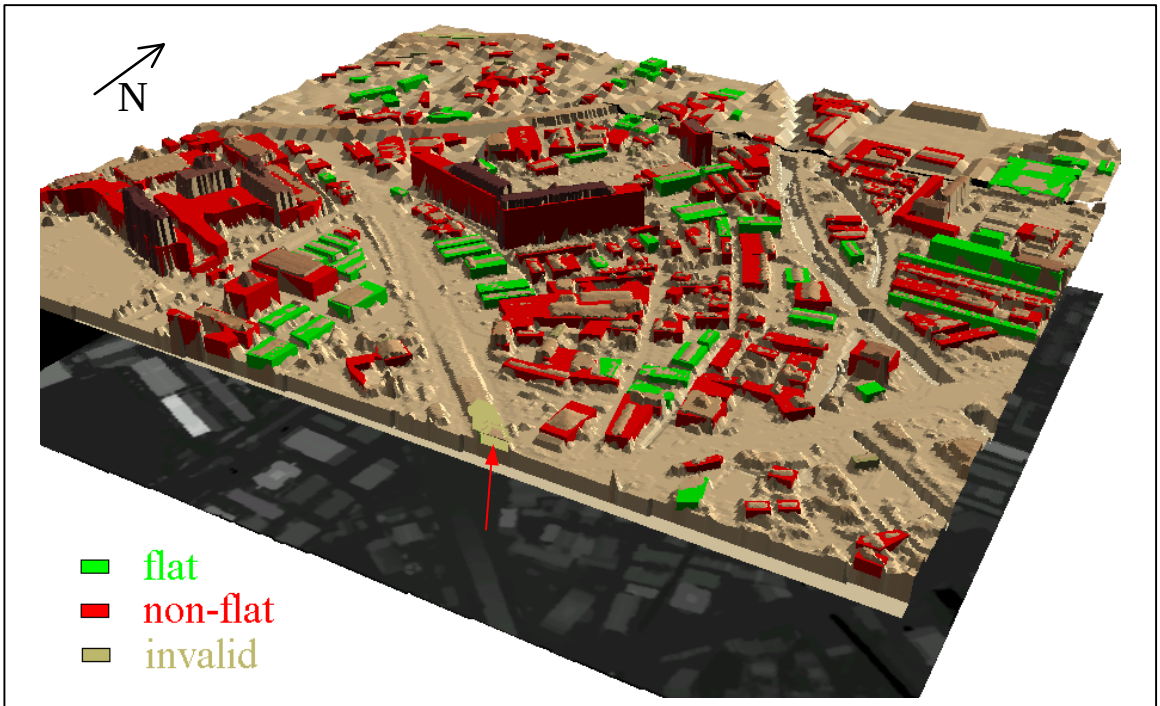
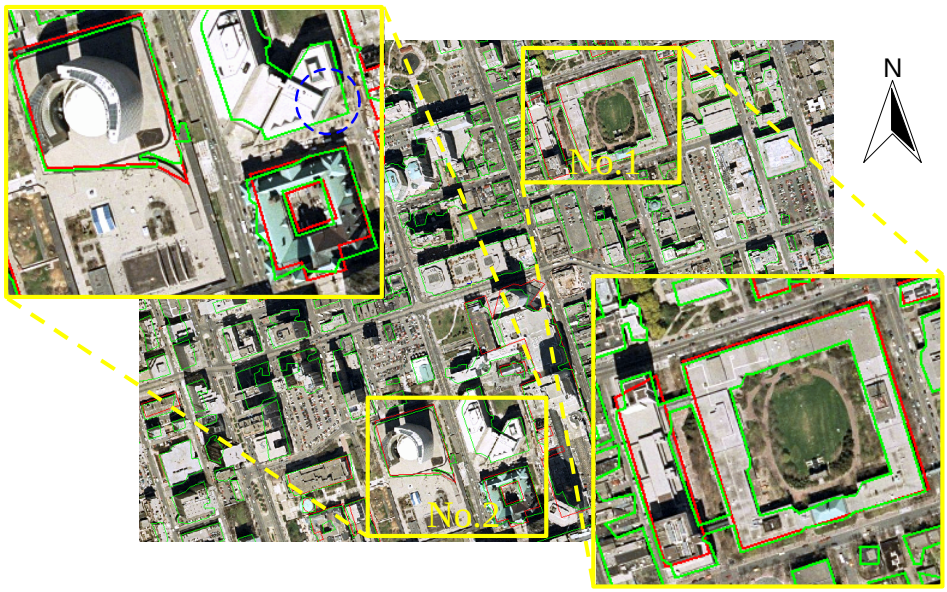
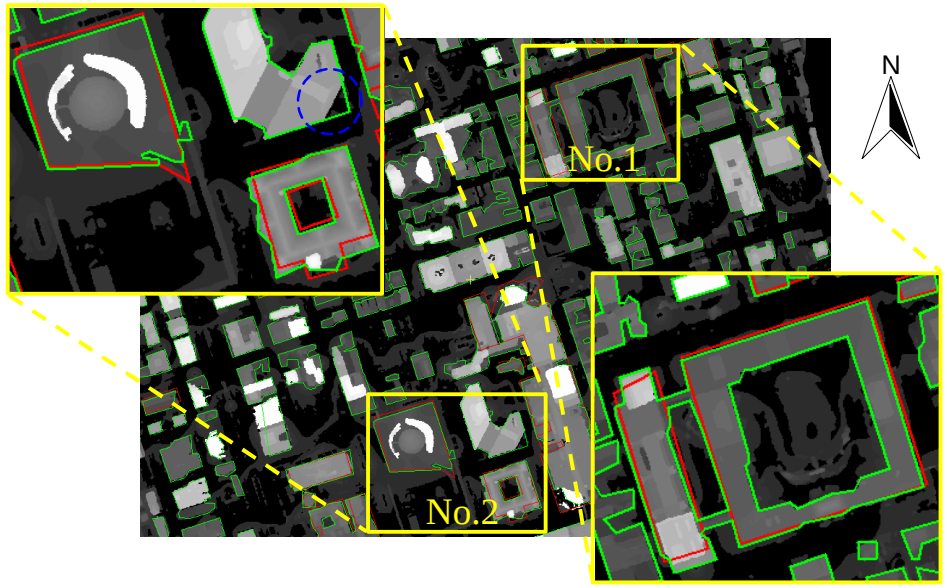


Figure 5.13. 3-D building models (Osaka city)



(a) Building footprint polygons overlaid on DOQs



(b) Building footprint polygons overlaid on the DNM

Figure 5.14. Comparison between detected and surveyed polygons

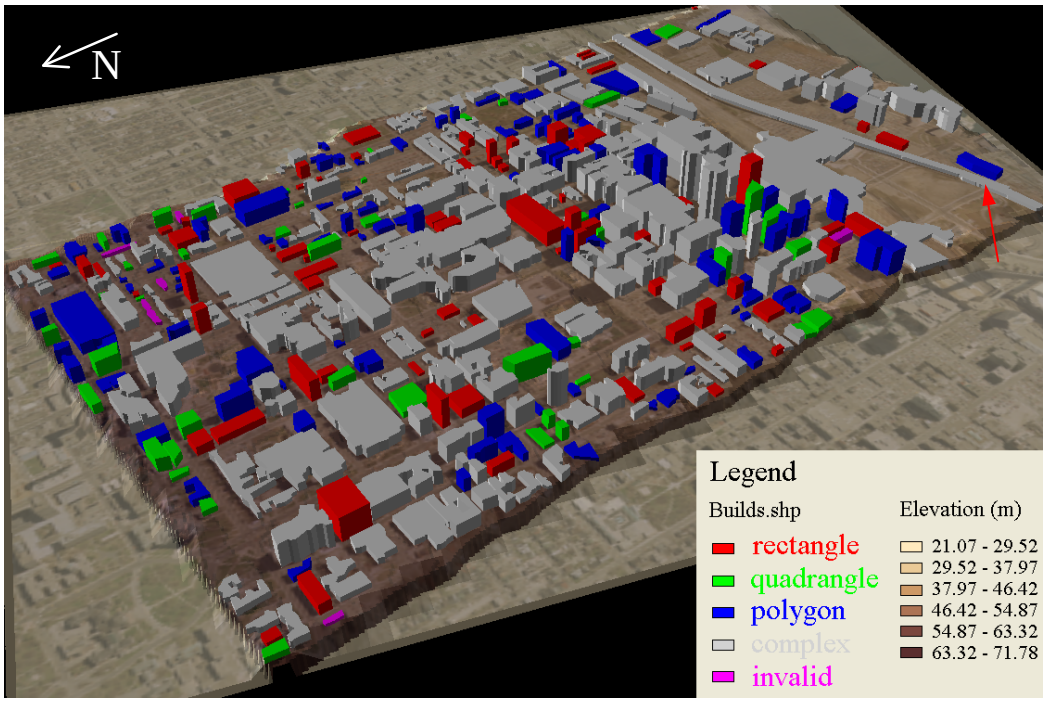


Figure 5.15. 3-D building models (Downtown Toronto)

Table 5.1. System performance evaluation

Stage	Performance indexes		Village site	Osaka city	Downtown Toronto
Building detection	Quantities	TP	33	170	403
		FP	0	16	16
		FN	1	> 30	> 33
	Quality measures (%)	Completeness	97	< 85	< 92
		Correctness	100	91	96
Building reconstruction	Quantities	TP	33	168	402
		FP	0	15	11
		FN	1	> 32	> 34
	Quality measures (%)	Completeness	97	< 84	< 92
		Correctness	100	92	97

5.9.2 Performance Evaluation

Table 5.1 gives the quantities and quality measures (see Section B.3) for above building extraction results. Three men [10] helped me to visually compare these building models with their corresponding patterns in DOQs or lidar images. The rule for determining a match is that the polygon model has an overlap larger than about 75% with the observed building patterns. A building model is considered to be detected correctly if it gets two or three votes. The buildings sited on the boundaries of the study areas are not counted since it is difficult to detect them due to the shortage of complete information.

Table 5.2. Statistical data of building extraction results

Quantities		Village site	Osaka city	Downtown Toronto
Observed buildings		34	> 200	> 436
Building detection	Buildings by classification	33	186	419
	Rectangles by searching	24 / 21	99 / 88	87 / 82
	Quadrangles by linking	1	13	52
	Polygons by linking	4	24	111
	Polygons by simplification	5	50	169
Building reconstruction	Final building models	33	183	413
	Prismatic models	0 / 0	33 / 32	14 / 14
	Polyhedral models	33 / 33	150	399
Attributes of reconstructed Buildings	Area range (m^2)	[72, 298]	[82, 9944]	[81, 93898]
	Average area (m^2)	154	598	3244
	Total area (m^2)	5083	109376	1339605
	Average height range (m)	[4, 7.6]	[3.7, 32.7]	[3.4, 175]
	95% maximal height range (m)	[5, 9.4]	[5.2, 46.7]	[6, 297]
	Boundary perimeter range (m)	[33, 83]	[37, 913]	[39, 3304]
	Boundary vertex number range	[4, 8]	[4, 54]	[4, 89]

[10] They are Mr. Jeff Xu (M.Sc. student in Photogrammetry), Mr. Ming Zhong (Undergraduate student in Computer Science), and Mr. Chris Wang (M.Sc. student in GIS), from York University.

Table 5.2 summarizes detailed statistical data about individual algorithms composing the overall methodology for above building extraction results. Some real situations have been observed from lidar DSMs and are listed as the denominators.

The quality of the extracted buildings and the proposed algorithms are assessed from following two aspects:

Quality analysis

- *Village site* (Figure 5.12). The building extraction results are very good since this scene is of low complexity. The completeness and correctness are as high as 97% and 100%, respectively.
  - No vegetation objects are classified as buildings. The vegetated objects are identified correctly either by shape measures or by the removal of small objects.
  - One house as indicated by the arrow is not detected. A part of its roof is occluded by adjacent trees. This corruption cannot be eliminated because only single-return data is available.

- *Osaka city* (Figure 5.13). Most buildings are correctly detected partly because tall trees seldom stand by them closely. The completeness and correctness are 84% and 92%, respectively.
  - The shortage of multi-return information leads to lower detection performance. About 11 vegetation-corrupted buildings are misclassified as vegetation.
  - About 19 building are missed because their areas are smaller than the area threshold (70 m^2). If I decrease the threshold, the correctness will decrease too.
  - Some small and low buildings are closely located with trees between them, and they are discriminated as a single building. This is one reason why the number of detected buildings is less than the observed number.
  - About 16 vegetated objects are misclassified as buildings. This show that the shape measures need to be improved or ancillary data have to be used.

- *Downtown Toronto* (Figures 5.14 and 5.15). The majority of buildings are detected correctly. The completeness and correctness are 92% and 97%, respectively.
  - It is observed that the reconstructed building boundaries are close to those obtained by filed surveyed in 2-D position (see Figure 5.14). Many reconstructed polygons give a better delineation of building boundaries than surveyed ones as enclosed in the first rectangle. But some reconstructed polygons are distorted a bit as enclosed in the second rectangle.
  - About 25 buildings with very complex boundaries are misclassified as vegetated regions. It is difficult to eliminate this misclassification by improving shape measures. Ancillary data have to be used.
  - Some thin strips of trees alongside streets are misclassified as buildings because their footprints have the rectangular shape. They are not removed even if the first and last returns are used. These thin tree strips are wrongly removed during the generation of the VSM because they are several meters wide and are very similar to the thin strips occurred at building boundaries (see Section 3.5.2).
  - Several tall buildings are detected as on melt building because they are connected via bridges larger than the segmentation threshold. Again, this is one reason why the number of detected buildings is less than the observed number.

Algorithm analysis

The performance of the methodology can be analyzed based on its component algorithms.

- The building boundary detection can reliably locate building footprints and express them as oriented boundaries by combining simple algorithms (see Figure 5.2).
  - The DNM excludes negative effects of terrain relief in a large scene, and put both buildings and vegetation objects on a flat reference plane. Object footprints are located first by a segmentation of the DNM and later by a connected component analysis. Processing the DNM has both advantages and disadvantages over processing the lidar DSM:

  * *Advantages*. The object boundaries are detected more reliably and efficiently using footprints rather than the DSM. The architectural details on building roofs and tree canopies in the lidar DSM will result in many disordered edges, and it is more difficult to reliably find the positions of their boundaries.
  * *Disadvantages*. Since the detailed edges are lost in the segmented DNM, several buildings may be recognized as one melt building if they are connected via bridges at the height larger than the segmentation threshold even the bridges are significantly lower than building roofs. If a larger height threshold is used, then some low buildings will be missed. A way is to slice the DNM and analyse the hierarchy of those segments, but has not tested.

  In addition, the representation of a boundary as a chain of oriented pixels is very useful for subsequent footprint reconstruction.
- Buildings are distinguished from vegetated regions using the shape measures based object classification scheme successful to some extent. But this may misclassify some buildings as vegetation objects when solely using single-return range data especially at forested areas (see the green polygon in Figure 5.5a). This is because tall trees may occlude low buildings and houses and result in distorted object footprints. This problem has been solved successfully by using the VSM.
- A difficulty encountered in the shape classification is the incapability of distinguishing buildings from aboveground long strip objects (e.g., bridges and viaducts as indicated by red arrows in Figures 5.13 and 5.15). To conquer this problem, extra geometric constraints or extra data have to be used.
- Most building footprints can be detected, and the completeness measures are 97%, 85% and 92%, respectively in Village site, Osaka and downtown Toronto datasets. Also, most building footprints are correctly detected, and the correctness measures are 100%, 91% and 96%, respectively in these three datasets. It is should be noted that the vegetation density is smaller in Osaka dataset than in downtown Toronto dataset. But the quality measures are larger for downtown Toronto than for Osaka. This is due the use of the VSM for downtown Toronto since that dataset has first and last returns. This shows that the use of the VSM can reliably eliminate vegetated regions, and thus improve the quality of building

detection in a typical scene with trees.

- The building boundary reconstruction can robustly transform complex bounding boundaries into rectangle, quadrangles or regular polygons using the proposed algorithms, including the constrained searching, the EHT and the SLT, and following the hypothesis verification paradigm and an optional refinement.
  - This workflow is data driven because it is not based on specific models, rather than complex shapes at the beginning. The rectangles, quadrangles and regular polygons are found step by step. The geometric regularity constraints are applied at last to refine the found polygons.
  - The constrained searching technique can detect rectangles reliably and robustly (see Figures 5.3 and 5.5a). This technique does not miss any rectangles for three datasets. But it accepts some non-rectangular shapes as rectangles, and the numbers are 3, 11 and 5, respectively, in Village, Osaka and downtown Toronto datasets. The misclassification rates are 14.3%, 12.5% and 6.1%, respectively. Downtown Toronto has obtained best result. This is because the multi-return information has helped to remove boundary corruptions from surrounding trees.
  - The EHT is composed of three processing steps, including grouping, merging and splitting. It can robustly and reliably divide a building boundary into a set of edge segments each associated with a group of successive edge points (see Figure 5.4b). The EHT improves the conventional HT and the SHT in three aspects.
    * First, the introduction of edge direction information improves the reliability when grouping line points compared to the SHT, and also reduces the computational burden by limiting the angle parameter to a sub-range around the edge direction. The precise grouping provides the possibility to permute edge segments according to distance measures.
    * Second, the merging operation effectively solves the problem of over-dividing a boundary based on the angle and distance evidences.
    * Third, the splitting operation successfully finds multiple separate edge segments along a straight line based on the connectivity information contained in the pixel chains.

- The SLT can reliably reconstruct polygonal boundaries based on hypothesis verification (see Figure 5.4c). It input the edge segments from the EHT, and link them in successive order to form the polygon. A hypothesized polygon is verified using multiple 2-D evidences about its simplicity and integrity. This procedure is very reliable, and only result in mis-linkage occasionally as enclosed in the circle in Figure 5.5e.
- The boundary reconstruction rates by constrained searching and sequential linking are 88%, 73% and 60%, respectively, in Village site, Osaka city and downtown Toronto datasets; and remaining building footprints are represented by simplified polygons. This is reasonable since Village site is simplest in content and downtown Toronto is most complex as observed from Figures 3.10 to 3.12.
- The boundary rectification systematically applies geometric regularity constraints about typical building shapes to individual U- and L-structures of reconstructed polygon model, and has achieved cartographical quality in many cases. As observed in Figure 5.5, the parallel edges and right corners are well modeled. But these regularity constraints may result in wrong re-shaping as enclosed in the circles in Figure 5.14.
- A footprint polygon can be verified using 1-D and 2-D evidences such as the allowable ranges of building attributes (see Table 5.2). The derived DTMs also provide valuable information for reconstructed polygon models since the elevations of their vertices can be retrieved from the DTM. This information is valuable because new constraints can be developed. That is, the maximum elevation difference of a footprint should be less than some height threshold.

• The building reconstruction can create prismatic models for flat roof buildings (see Figure 5.8) and polyhedral models for non-flat roof buildings (see Figure 5.9).
  - The selective refinement technique can approximate a complex building roof efficiently using a minimum set of most significant roof points. The vertical wall constraint has been used to identify three types of ridge vertices in the polyhedral model to produce a more regular shape. But some corner points are not well rectified yet (see Figure 5.11).

- The statistical testing is efficient for classifying flat roofs from non-flat roofs because lidar points often have a homogenous spacing on ground. No flat roofs are discriminated as non-flat ones, and only one non-flat roof is misclassified to be flat in all the three datasets (see Table 5.2).
- A building model can be verified by examining 3-D evidences such as the existence of vertical walls. The correctness slightly increases from 91% to 92% in Osaka dataset, and from 96% to 97% in downtown Toronto dataset. The completeness for downtown Toronto keeps same, and is 92%. While the completeness for Osaka slightly decreased from 85% to 84% since two buildings are discarded wrongly at this stage. This also shows the high reliability of the building detection algorithms.

• The quality evaluation for automatically extracted buildings is seldom discussed or even omitted in the literature. Most researchers given emphases on their algorithms only, and reported their results using small datasets, showing a few separate or small groups of buildings. In addition, they have processed scenes with only specific types of buildings such as rectangle (Weidner, 1995; Jaynes et al., 1997; Wang and Schenk, 2000; Gamba and Houshmand, 2002), or used ground plans (Haala et al., 1998). To my knowledge, no other algorithms in the literature have presented the similar capability of automated reconstruction of complex footprints.

5.10 SUMMARY

I developed a data-driven methodology for automated extraction of buildings using lidar height data and multi-return information. The proposed algorithms have produced good results when testing several lidar datasets with varying complexities and manually checking the extracted results against ground observations.

Using height data, building detection turns into locating footprints of objects that are above the terrain surface and have regular shapes. Using the VSM, the footprints of most non-building objects can be reliably identified and deleted. The object boundaries are

expressed as closed chains of oriented edge pixels. The candidate building boundaries are distinguished from non-building ones by examining their shape measures. Rectangular boundaries are reconstructed by employing a constrained searching in Hough space that is very robust. If an oriented boundary does not have the rectangular shape, it is robustly divided into edge segments each with a group of edge pixels by applying the EHT. These edge segments are further permuted in correct order using the SLT to form a closed polygon. Some geometric regularity constrains are then applied to produce regular polygonal shapes for buildings.

The average roof height of a candidate building is computed by collecting lidar points falling inside its polygonal shape, and prismatic models are created for flat roof buildings. Complex roof buildings are modeled as polyhedral shapes, which are reconstructed by a selective refining procedure and are rectified by applying the vertical wall constraint. This refining procedure selects and adds most significant roof points iteratively to obtain the best roof shape. A candidate building may be discarded by examining multiple criteria such as point density and roof height at this stage. The major properties about each building, including its area, main orientation etc., are calculated.

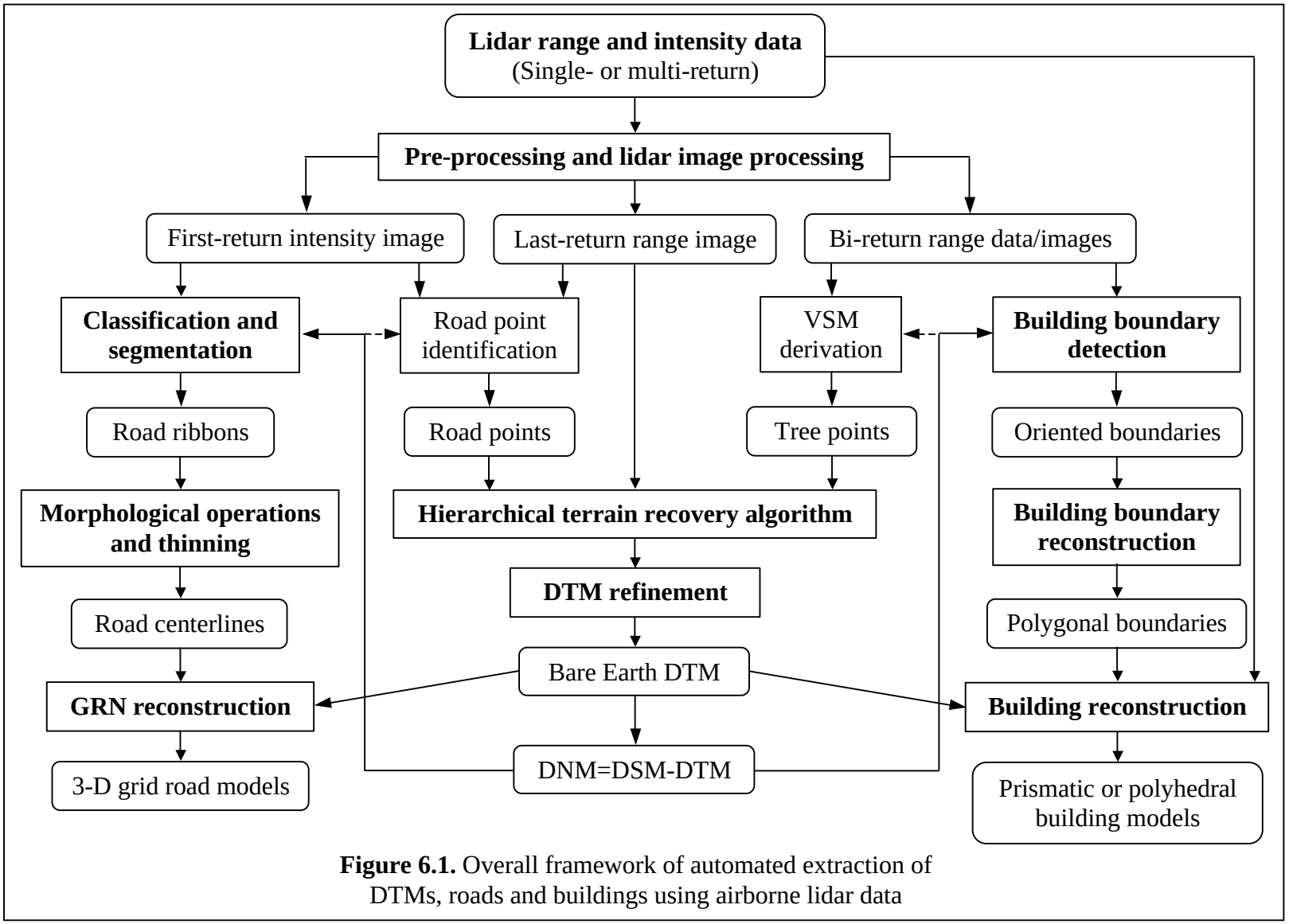


Figure 6.1. Overall framework of automated extraction of DTMs, roads and buildings using airborne lidar data

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

The algorithms developed in Chapters 3 to 5 have implemented automated, robust and efficient information extraction using single- or multi-return lidar range and intensity data. Figure 6.1 gives the overall framework of the proposed processes for automated extraction of DTMs, roads and buildings from airborne lidar data. The individual processes and their interrelations have been described and evaluated in previous chapters. The integration of range and intensity data makes use of geometry and radiometry information about ground features, and thus provides the possibility for understanding complex scenes. The employment of effective processing strategies has improved the automation capabilities in feature extraction from lidar data. Following conclusions can be drawn from this research:

- *DTM generation*. An efficient framework (see the middle part of Figure 6.1) has been developed to generate DTMs automatically for single- or multi-return range and intensity data in large datasets.
  - The HTRA is data driven, and does not assume a priori knowledge about the scene complexity. It is suited to process complex scenes sampled in different densities under a uniform technical framework.
  - The success of the algorithm is achieved by the effective combination of multiple component techniques including the hierarchical approach, smooth condition, data fusion, and interpolation methods. The success of the algorithm is also verified by the success of the automated extraction of roads and buildings, which needs height information derived by it.
  - The algorithm is based on the assumption: local lowest points in moving windows are sited on the bare Earth. To ensure the success generation of DTMs, the largest window size should be larger than the smaller dimension of any non-terrain object,

and the lidar data should be acquired in the leaf-off condition at forested areas.

- The worst-case relative accuracy, at the confidence level of 90%, is better than 25 cm at building and tree surroundings and bare surfaces for datasets acquired at residential, suburban, urban, forested or mountainous areas.

- *Road extraction*. New algorithms (see the left part of Figure 6.1) have been developed for automated detection of road ribbons for complex scenes and reconstruction of 3-D grid road networks in built-up areas using multi-return intensity and range data.
  - The integration of both the radiometry property of intensity data and the height information of the digital non-terrain model is highly feasible in reducing the misclassification of the road class.
  - The resulting road ribbons are not perfectly complete due to three reasons: the new pavement with concrete, the undetected bridges and viaduct, and different material other than asphalt. They may contain non-road areas, including outdoor parking lots and some water areas due to similar reflection rate as asphalt and nearly zero height as roads.
  - The reconstruction of grid road networks in urban areas is model driven, and is based on the global grid constraint. Therefore, the premise is the automatic detection of areas with grid networks.
  - The road segment based hypothesis and verification strategy leads to efficient, robust, and automatic reconstruction of 3-D grid road models. The algorithm is robust to some irregular road patterns in the road network.
  - The grid road models match the grid road networks well in urban areas with regular street grids. The road segments and crossings have the completeness of 84% and 83%, respectively, and the correctness of 91.6% and 78%, respectively, in downtown Toronto dataset. The redundant and undetected road segments are relatively shorter than correctly detected ones, and correctly detected road crossings are located accurately.

- *Building extraction*. New algorithms (see the right part of Figure 6.1) have been developed to implement a data driven methodology for automated extraction of

buildings using lidar height data and multi-return information in large datasets.

- The boundary detection can reliably locate buildings from the height data and the vegetation support model. The representation of building boundaries using oriented boundaries is very useful for subsequent boundary reconstruction. The building detection rates are 97%, 85% and 92%, respectively in Village site, Osaka and downtown Toronto datasets.
- The reconstruction of building boundaries is data driven, and is not based on specific footprint models, but rather on complex footprint shapes. This processing can robustly transform complex building boundaries as rectangle, quadrangles or regular polygons using the constrained searching and the sequential linking technique and following the hypothesis verification paradigm and further refinements. The boundary reconstruction rates by constrained searching and sequential linking are 88%, 73% and 60%, respectively in Village site, Osaka city and downtown Toronto datasets.
- The building reconstruction can create prismatic models for flat roof buildings and polyhedral models for non-flat roof buildings. The selective refinement technique can approximate complex building roof efficiently using a minimum set of most significant roof points. The vertical wall constraint can be used to identify three types of ridge vertices in the polyhedral model to obtain a better shape.

6.2 RECOMMENDATIONS FOR FUTURE WORK

This work also unveils a number of new topics worth of further research efforts. The recommended research contents are addressed below. Solving these problems will make the information extraction from lidar data more practical for industry production.

- I could manually label some terrain and non-terrain points in a lidar dataset before processing them using the algorithm. Fusing ancillary data including the GIS data and multi-spectral images also could largely eliminate the uncertainty in identifying road or terrain points. For example, to avoid smoothing abrupt terrain changes at cliffs, and dams, I could incorporate breaklines and mass points from photogrammetry or GPS

for DTM generation. The problem is how to efficiently incorporate additional information into the DTM generation procedure. In addition, I could directly use the 3-D points instead of creating and processing rasterized images. This may make the facet models and slope estimates better approximate the local geomorphology and retain accurate details when performing interpolation. To process extremely large datasets that cannot be processed in the memory as a whole, a solution is to divide the whole scene into multiple overlapping sections, and the individual processing results are then merged.

- The road ribbons or rough centerlines produced from lidar data could be useful cues to guide road extraction in high-resolution optical imagery. The rough centerlines are expected to be able to be refined by analyzing the standard road crossing models. In a large dataset covering areas with grid and non-grid road networks, the automatic determination of grid areas is also needed before applying the proposed algorithm.

- The difficulty of the building extraction task could be greatly reduced when utilizing application specific assumptions. For example, the model-based recognition is feasible if the region of interest has only buildings with rectangular footprints and flat or gable roofs. Building detection could be enhanced by classifying lidar intensity data and multi-spectral images. In the rectification of polyhedral models using the vertical wall constraint, further refinement is to correct distortions around corners using the geometric regularity constraints of footprints. I am also interested in modeling compound building boundaries with inner structures.

REFERENCES

Acronym	Source title
AEMOASI	Automatic Extraction of Manmade Objects from Arial and Space Images
CVGIP	Computer Vision, Graphics and Image Processing
CVPR	Computer Vision and Pattern Recognition
IAPRS	International Archives of Photogrammetry and Remote Sensing
ICPR	International Conference on Pattern Recognition
JPRS	Journal of Photogrammetry and Remote Sensing
PAMI	Pattern Analysis and Machine Intelligence
PE&RS	Photogrammetric Engineering and Remote Sensing

Ackermanm, F., 1999. Airborne laser scanning - present status and future expectations, *ISPRS JPRS*, 54: 64-67.

Airault, S., Jamet, O., Leymarie, F., 1996. From manual to automatic stereo plotting: evaluation of different road network capture process, *IAPRS*, vol. 31, pp. 14-18.

Airbornelasermapping (ALM), 2003. Airborne laser mapping: a reference on an emerging technology, URL: http://Airbornelasermapping.com/.

Alharthy, A., Bethel, J., 2002. Building extraction and reconstruction from lidar data, *ACSM-ASPRS Annual Conference*, 19-26 April, Washington DC, 8 p.

Alharthy, A., Bethel, J., 2003. Automated road extraction from lidar data, *ASPRS Annual Conference*, 3-9 May, Anchorage, AK, 8 p.

Ameri, B., 2000. Feature based model verification: a new concept for hypothesis validation in building reconstruction, *IAPRS*, Amsterdam, vol. 33, part B3, pp. 24-35.

Arce, G., and Forster, R., 1989. Detail preserving ranked-order based filters for image processing, *IEEE Trans. on Acoustics, Speech and Signal Processing*, 37(1): 83-98.

Axelsson, P., 1999. Processing of laser scanner data - algorithms and applications, *ISPRS JPRS*, 54: 138-147.

Axelsson, P., 2000. DEM generation from laser scanner data using adaptive TIN models,

IAPRS, 17-22 July, Amsterdam, vol. 33, part B4, pp. 110-117.

Baltsavias, E.P., 1999a. Airborne laser scanning: existing systems and firms and other resources, *ISPRS JPRS*, 54: 164-198.

Baltsavias, E.P., 1999b. Airborne laser scanning - relations and formulas, *ISPRS JPRS*, 54: 199-214.

Baltsavias, E.P., Mason, S., Stallmann, D., 1995. Use of DTMs/DSMs and orthoimages to support building extraction, *Workshop on AEMOASI*, Basel, pp. 199-210.

Baruch, O., Loew, M.H., 1988. Segmentation of two-dimensional boundaries using the chain code, *Pattern Recognition*, 21(6): 581-589.

Barzohar, M., Cooper, D.B., 1996. Automatic finding of main roads in aerial images by using geometric-stochastic models and estimate, *IEEE T-PAMI*, 18(7): 707-721.

Baumgartner, A., Steger, C., Mayer, H., Eckstein, W., 1997. Multi-resolution, semantic objects and context for road extraction, *Workshop on Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*, Basel, pp. 140-156.

Bergholm, F., 1987. Edge focusing, *IEEE Trans. on PAMI*, 9(6): 726-741.

Besl, P.J., 1988. *Surfaces in Range Image Understanding*, Springer-Verlag, New York, 339 p.

Besl, P.J., Jain, R.C., 1988. Segmentation through variable-order surface fitting, *IEEE Trans. on PAMI*, 10(2): 167-192.

Brenner, C., 1999. Interactive modeling tools for 3D building reconstruction, *Photogrammetric Week*, Heidelberg, pp. 23-34.

Brenner, C., 2000. Towards fully automatic generation of city models, *IAPRS*, 17-22 July, Amsterdam, vol. 33, part B3, pp. 85-92.

Brovelli, M.A., Cannata, M., Longoni, U.M., 2002. Managing and processing LIDAR data within GRASS, *Proceedings of the Open Source GIS - GRASS Users Conference*, 11-13 September, Trento, Italy, 29 p.

Brunn, A., Weidner, U., 1998. Hierarchical Bayesian nets for building extraction using dense digital surface models, *ISPRS JPRS*, 53: 296-307.

Bureau of Transportation Statistics (BTS), 2002. Strategies for creating a national pipeline mapping system, URL: http://www.bts.gov/gis/reference/npms_content.html.

Burtch, R., 2002. Lidar principles and applications, *IMAGIN Conference: Geography on*

the Move, 29 April - 1 May, Travers, MI, 13 p.

Canny, J., 1986. A computational approach to edge detection, *IEEE Trans. on PAMI*, 8(6): 769-797.

City of Bellingham (COB), 1999. Whatcom creek pipeline incident: maps & photos of Whatcom creek area, Washington, URL: http://www.cob.org/whatcomcreek.htm.

Coleman, D., 2001. Radar revolution: revealing the bald earth, *Earth Observation Magazine*, 10(11). URL: http://www.eomonline.com/.

Couloigner, I., Ranchin, T., 2000. Mapping of urban areas: a multi-resolution modeling approach for semi-automatic extraction of streets, *PE&RS*, 66(7): 867-874.

De Berg, M., Kreveld, M., Overmars, M., Schwarzkopf, O., 2000. *Computational Geometry: Algorithms and Applications*, New York: Springer, 367 p.

Duda, R.O., Hart. P.E., 1972. Use of the Hough transformation to detect lines and curves in pictures, *Communications of the ACM*, 15(1): 11-15.

Egels, Y., Kasser, M., 2001. *Digital Photogrammetry*, Taylor & Francis, New York, 304 p.

Elaksher, A.F., Bethel, J.S., 2002. Building extraction using lidar data, *ACSM-ASPRS Annual Conference*, 19-26 April, Washington DC, 9 p.

Elberink, S.O., Mass, H., 2000. The use of anisotropic height texture measures for the segmentation of airborne laser scanner data, *IAPRS*, 17-22 July, Amsterdam, vol. 33, part B3, pp. 678-684.

Elmqvist, M., 2001. Ground surface estimation from airborne laser scanner data using active shape models, *OEEPE Workshop on Airborne Laserscanning and IfSAR for Detailed DEMs*, 1-3 March, Stockholm, 5 p.

El-Sheimy, N., 1996. *The Development of VISAT - A Mobile Survey System for GIS Applications* [Ph.D. thesis], UCGE Report No. 20101, Department of Geomatics Engineering, University of Calgary, 175 p.

El-Sheimy, N., 1998. *Digital Terrain Modeling*, ENGO 573, Department of Geomatics Engineering, University of Calgary, URL: http://www.geomatics.ucalgary.ca/%7Enel-shei/engo573.htm.

Fischler, M.A., Heller, A.J., 1998. Automated techniques for road network modeling, *DARPA Image Understanding Workshop*, Monterey, CA, pp. 501-516.

Fischler, M.A., Tenenbaum, J.M., Wolf, H.C., 1981. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique, *Computer Graphics & Image Processing*, 15: 201-223.

Flood, M., 2001. Laser altimetry: from science to commercial lidar mapping, *PE&RS*, 67(11): 1209-1218.

Flood, M., 2002. Product definitions and guidelines for use in specifying lidar deliverable, *PE&RS*, 68(12): 1230-1236.

Fowler, R., 2001. Chapter 7: Topographic Lidar, in: David, F.M. (editor), 2001. *Digital Elevation Model Technologies and Applications: The DEM Users Manual*, Bethesda, Maryland: ASPRS, pp. 207-236.

Fraser, C., Jonas, D., 2001. Report on 1998 airborne laser scanner trials (Version 2), AAM Surveys Pty Limited, 42 p.

Gamba, P., Houshmand, B., 2002. Joint analysis of SAR, LIDAR and aerial imagery for simultaneous extraction of land cover, DTM and 3D shape of buildings, *International Journal of Remote Sensing*, 23(20): 4439-4450.

Garnesson, P., Giraudon, G., Montesinos, P., 1990. An image analysis system: application for aerial imagery interpretation, *International Conference on Pattern Recognition*, 16-21 June, Atlantic, vol. 1, pp. 210-212.

Geibel, R., Stilla, U., 2000. Segmentation of laser altimeter data for building reconstruction: different procedures and comparison, *IAPRS*, 17-22 July, Amsterdam, vol. 33, part B3, pp. 326-334.

Geiger, D., Yuille, A., 1991. A common framework for image segmentation, *International Journal of Computer Vision*, 24(6): 227-243.

Glasbey, C.A., 1993. An analysis of histogram-based thresholding algorithms, *CVGIP: Graphical Models and Image Processing*, 55(6): 532-537.

Gonzalez, R.C., Woods, R.E., 1992. *Digital Image Processing*, Addison-Wesley, New York, 716 p.

Greve, C.W., 1997. *Digital Photogrammetry: An Addendum to the Manual of Photogrammetry*, ASPRS, Bethesda, 250 p.

Gruen, A., Li, H., 1994. Semi-automatic road extraction by dynamic programming, *IAPRS*, pp. 324-332.

Gruen, A., Li, H., 1997. Linear feature extraction with 3-D LSB-Snakes, *Workshop on AEMOASI*, pp. 287-297.

Haala, N., Anders, K.H., 1997. Acquisition of 3-D urban models by analysis of aerial images, digital surface models and existing 2-D building information, *SPIE Conference on Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*, Orlando, pp. 212-222.

Haala, N., Brenner, C., 1999. Extraction of buildings and trees in urban environments, *ISPRS JPRS*, 54: 130-137.

Haala, N., Brenner, C., Anders, K.H., 1998. 3-D urban GIS from laser altimeter and 2-D map data, *IAPRS*, vol. 32, pp. 339-346.

Hazarika, M.K., Honda, K., Samarakoon, L., Murai S., 1999. Application of remote sensing for extraction of road information, *The 20th Asian Conference on Remote Sensing*, 22-25 November, Hong Kong.

Haugerud, R.A., Harding, D.J., 2001. Some algorithms for virtual deforestation of lidar topographic survey data, *IAPRS*, 22-24 October, Annapolis, pp. 211-218.

Heckbert, P.S., Garland, M., 1997. Survey of polygonal surface simplification algorithms, Technical Report, Computer Science Department, Carnegie Mellon University, 29 p.

Hershberger, J., Snoeyink, J., 1992. Speeding up the Douglas-Peucker line simplification algorithm, *International Symposium on Spatial Data Handling*, August, Charleston, vol. 1, pp. 134-143.

Henricson, O., Bignone, F., Willuhn, W., Ade, F., Kubler, O., Baltsavias, E., Mason, S., Gruen, A., 1996. Project Amobe: strategies, current status and future work, *IAPRS*, vol. 31, pp. 321-330.

Hill, J.M., Graham, L.A., Henry, R.J., 2000. Wide-area topographic mapping and applications using light detection and ranging technology, *PE&RS*, 66, 908-914.

Hinz, S., Baumgartner, A., 2000. Road extraction in urban areas supported by context objects, *IAPRS*, vol. 33, part B3, pp. 405-412.

Hofmann, P., 2001. Detecting buildings and roads from IKONOS data using additional elevation information, *GIS*, 6: 28-33.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P.J., Bunke H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., Fisher, R.B., 1996. An experimental comparison of

range image segmentation algorithms, *IEEE Trans. on PAMI*, 18(7): 673-689.

Hu, X., Tao, V., 2003a. A robust method for interactive extraction of road centerlines from spatial imagery, *PE&RS* (accepted).

Hu, X., Tao, V., 2003b. Automatic extraction of main-road centerlines from high resolution satellite imagery based on perceptual grouping, *ASPRS Annual Conference*, 3-9 May, Anchorage, AK, 9 p.

Hu, Y., Tao, V., 2002. Bald DEM generation and building extraction using range and reflectance lidar data, *ACSM-ASPRS Annual Conference and FIG XXII International Congress*, 19-26 April, Washington DC, 11 p.

Hu, Y., Tao, V., 2003. Automatic extraction of digital terrain models and road networks using multiple returns lidar data, *ASPRS Annual Conference*, 3-9 May, Anchorage, AK, 11 p.

Hu, Y., Tao, V., Collins, M., 2003. Automatic extraction of buildings and generation of 3-D city models from airborne lidar data, *ASPRS Annual Conference*, 3-9 May, Anchorage, AK, 12 p.

Hug, C., 1997. Extracting artificial surface objects from airborne laser scanner data, *Workshop on AEMOASI*, Basel, pp. 203-212.

Hug, C., Wehr, A., 1997. Detecting and identifying topographic objects in imaging laser altimeter data, *IAPRS*, vol. 32, pp. 19-26.

ISM, 1997. *The Essential Photogrammetric Handbook*, International Systemap Corp., URL: http://www.ismcorp.com/index.asp?page=booklet.

Jaynes, C.O., Hanson, A., Rieseman, E., Schultz, H., 1997. Building reconstruction from optical and range images, *CVPR*, University of Massachusetts, 7 p.

Jian, Y., Hu, Y., Li, J., Sun, Z., 1998. Image matching technology based on Hausdorff distance (in Chinese), *Infrared and Laser Engineering*, 27(4).

Kaneko, T., Okudaira, M., 1985. Encoding of arbitrary curves based on the chain code representation, *IEEE Trans. on Communications*, 33(7): 697-707.

Kass, M., Witkin, A., Terzopoulos, D., 1987. Snakes: active contour models, *International Journal of Computer Vision*, 1(4): 321-331.

Kilian, J., Haala, N., Englich, M., 1996. Capturing and evaluation of airborne laser scanner data, *IAPRS*, Vienna, vol. 32, Part B3, pp. 383-388.

Kraus, K., Pfeifer, N., 1998. Determination of terrain models in wooded areas with ALS data, *ISPRS JPRS*, 53, 193-203.

Kraus, K., Pfeifer, N., 2001. Advanced DTM generation from lidar data, *IAPRS*, 22-24 October, Annapolis, Maryland, vol. XXXIV, part 3/W4, pp 23-30.

Kraus, K., Rieger, W., 2000. Processing of laser scanning data for wooded areas, *Photogrammetric Week*, Stuttgart, German, pp. 221-231.

Lacroix, V., 1990. The primary raster: a multi-resolution image description, *ICPR*, 16-21 June, Atlantic, pp. 903-907.

Laptev, I., 1997. *Road Extraction Based on Line Extraction and Snakes* [Mater Thesis], Royal Institute of Technology, Stockholm, Sweden, 71 p.

Lemmens, M., Deijkers, H., Looman, P., 1997. Building detection by fusing airborne laser altimeter DEMs and 2-D digital maps, *IAPRS*, vol. 32, pp. 42-49.

Li, J., 1999. *Theories and Practices in Computer Vision* (2nd version, in Chinese), Press of Shanghai Jiaotong University.

Lin, C., Nevatia, R., 1998. Building detection and description from a single intensity image, *Computer Vision & Image Understanding*, 72(2): 101-121.

Lischinski, D., 1994. Incremental Delaunay triangulation, In: Heckbert, P., editor, 1994. *Graphics Gems IV*, Academic Press, Boston, pp. 47-59.

Maas, H., Vosselman, G., 1999. Two algorithms for extracting building models from raw altimetry data, *ISPRS JPRS*, 54: 153-63.

Mallat, S., 1989. A theory for multiresolution signal decomposition: the wavelet decomposition, *IEEE Trans. on PAMI*, 11(7): 674-693.

Masaharu, H., Ohtsubo, K., 2002. A filtering method of airborne laser scanner data for complex terrain, *IAPRS*, 9-13 September, Graz, vol. 34, part 3A/B, 5 p.

Mayer, H., Laptev, I., Baumgartner, A., Steger, C., 1997. Automatic road extraction based on multi-scale modeling, context and snakes, *IAPRS*, vol. 32, part 3-2W3, pp. 106-113.

McIntosh, K., Krupnik, A., 2002. Integration of laser-derived DSMs and matched image edges for generating an accurate surface model, *ISPRS JPRS*, 56, 167-176.

McKeown, D.M., Denlinger, J.L., 1988. Cooperative methods for road tracing in aerial imagery, *CVPR*, 5-9 June, Ann Arbor, pp. 662-672.

Mercer, B., 2001. Combining LIDAR and IFSAR: what can you expect, *Photogrammetric Week*, Stuttgart, 11 p.

Migon, H.S., Garmerman, D., 1999. *Statistical Inference: An Integrated Approach*. Arnold, London, 262 p.

Mikhail, E.M., Bethel, J.S., McGlone, J.C., 2001. *Introduction to Modern Photogrammetry*, John Wiley & Sons, Inc., New York, 479 p.

Mohamed, A., 2003. Navigating the ground from air: active monitoring with GPS/INS geo-referenced lidar, *ION Conference*, 3 January, Anaheim, pp. 593-601.

Mohamed, A., Price, R., 2003. Near the speed of flight – aerial mapping with GPS/INS direct georeferencing, *GPS World*, 13(3): 40-45.

Mohamed, A., Price, R., McNabb, D., Green, J., Spence, P., 2001. The development of DORIS: an overview, *Int. Symposium on Mobile Mapping Technology*, 3-5 January, Cairo, Egypt, 11 p.

Molander, C., Merritt, S., Corrubia, A., 2002. Marrying photogrammetry and lidar, *Earth Observation Magazine*, 11(6). URL: http://www.eomonline.com/.

Murakami, H., Nakagawa, K., Hasegawa, H., Shibata, T., Iwanami, E., 1999. Change detection of buildings using an airborne laser scanner, *ISPRS JPRS*, 54, 148-152.

National Imagery Mapping Agency (NIMA), 2001. *Airfield Initiative Document*, 50 p. URL: http://www2.nima.mil/products/rbai/AIDOCwww.zip.

NPMS, 2002. *National Pipeline Mapping System*, URL: http://www.npms.rspa.dot.gov/.

National Aeronautics and Space Administration (NASA), 1999. *LIDAR Tutorial*, URL: http://www.ghcc.msfc.nasa.gov/sparcle/sparcle_tutorial.html.

Nevatia, R., Huertas, A., Kim, Z., 1999. The MURI project for rapid feature extraction in urban area, *IAPRS*, vol. 32, 12 p.

Office of Pipeline Safety (OPS), 2002. *Pipeline Statistics*, URL: http://ops.dot.gov/.

Optech, 2003. *Laser-Based Ranging, Mapping and Detection Systems*, URL: http://www.optech.on.ca/.

O'Rourke, J., 1998. *Computational Geometry in C* (2nd Edition), Cambridge University Press, Cambridge, UK, 368 p.

Papamarkos, N., Gatos, B., 1994. A new approach for multilevel threshold selection, *CVGIP: Graphical Models and Image Processing*, 56(5): 357-370.

Park, J.S., Saleh, R.A., 2001. A comprehensive survey of extracting techniques of linear features from remote sensing imagery, *Meeting on Road Centerline Extraction & Maintenance*, 6-7 August, Santa Barbara, URL: http://www.ncgia.ucsb.edu/ncrst/.

Parker, J.R., 1993. *Practical Computer Vision Using C*, John Wiley & Sons Inc., New York, 476 p.

PE&RS, 2001. Direct georeferencing column: an introduction, *PE&RS*, 67(10): 1105, 1107, 1109.

Petzold, B., Reiss, P., Stossel, W., 1999. Laser scanning - surveying and mapping agencies are using a new technique for the derivation of digital terrain models, *ISPRS JPRS*, 54, 95-104.

Pfeifer, N., Stadler, P., Briese, C., 2001. Deriving of digital terrain models in the SCOP++ environment, *OEEPE Workshop on Airborne Laserscanning and IfSAR for Detailed DEMs*, Stockholm, 13 p.

Pipe Line & Gas Industry (PLGI), 2002. *Archive*, URL: http://www.pipe-line.com/.

Pitas I., 2000. *Digital Image Processing Algorithms and Applications*, John Wiley & Sons, New York, 419 p. URL: http://ftp.wiley.com/public/sci_tech_med/.

Pitas, I., Venetsanopoulos, A.N., 1990. *Non-linear Digital Filters: Principles and Applications*, Kluwer Academic, 392 p.

Pratt, W.K., 2001. *Digital Image Processing* (Third Edition), John Wiley & Sons, Inc., New York, 735 p.

Price, K., 1999. Road grid extraction and verification, *IAPRS*, vol. 32, pp. 101-106.

Rabor, G.T., Jensen, J.R., Schill, S.R., and Schuckman, K., 2002. Creation of digital terrain models using an adaptive lidar vegetation point removal process, *PE&RS*, 68(12): 1307-1315.

Ritter, G.X., Wilson, J.N., 2000. *Handbook of Computer Vision Algorithms in Image Algebra* (2nd edition), CRC Press, Boca Raton, Florida, 384 p.

Rottensteiner, D.F., 2001. *Semi-automatic extraction of buildings based on hybrid adjustment using 3D surface models and management of building data in a TIS* [Ph.D. dissertation], Vienna University of Technology. URL: http://www.ipf.tuwien.ac.at/fr/.

Rottensteiner, D.F., Briese, C., 2002. A new method for building extraction in urban areas from high-resolution lidar data, *IAPRS*, 9-13 September, Graz, Austria, vol. 34,

part 3A/B, pp. 295-301.

Rucklidge, W.J., 1997. Efficiently locating objects using Hausdorff distance, *International Journal of Computer Vision*, 24(3): 251-270.

Ruskone, R., 1996. *Road Network Extraction by Local Context Interpretation: Application to the Production of Cartographic Data* [Ph.D. thesis], Marne-La-Vallee University, France.

Saaty, T.L., 1980. *The Analytical Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill International Book Co., New York, 287 p.

Saint-Marc, P., Chen, J., Medioni, G., 1991. Adaptive smoothing: a general tool for early vision, *IEEE Trans. on PAMI*, 13(6): 514-529.

Sapeta, K., 2000. Have you seen the light? LIDAR technology is creating believers, *GeoWorld*, 13(10): 32-36.

Satale, D.M., Kulkarni, M.N., 2003. Lidar in mapping, *Map India*, 28-31 January, New Delhi, India. URL: http://www.gisdevelopment.net/technology/gis/mi03129.htm.

Schickler, W., Thorpe, A., 2001. Surface estimation based on lidar, *ASPRS Annul Conference*, 23-27 April, St. Louis, 11 p.

Schiewe, J., 2003. Integration of multi-sensor data for landscape modeling using a region-based approach, *ISPRS JPRS*, 57: 371-379.

Sester, M., 2000. Generalization based on least squares adjustment, *IAPRS*, 17-22 July, Amsterdam, vol. 33, part B4, pp. 931-938.

Shen, Y., Liang, Z., Xu, L., Cai, L., 1992. *Practical Mathematics Handbook* (in Chinese), Scientific Press, Beijing, 1068 p.

Shirai, Y., 1987. *Three-Dimensional Computer Vision*, Springer-Verlag, Berlin, 297 p.

Sithole, G., 2001. Filtering of laser altimetry data using a slope adaptive filter, *IAPRS*, 22~25 October, Annapolis, Maryland, vol. 34, part 3/W4, pp. 203-210.

Sithole, G., 2002. Filtering strategy: working toward reliability, *IAPRS*, 9-13 September, Graz, Austria, vol. 34, part 3A/B, pp. 330-335.

Sithole, G., Vosselman, G., 2003. ISPRS test on extracting DEMs from point clouds: a comparison of existing automatic filters, Technical Report, 93 p.

Song, J.H., Han, S.H., Yu, K., Kim, Y., 2002. Assessing the possibility of land-cover classification using lidar intensity data, *IAPRS*, 9-13 September, Graz, vol. 34, 4 p.

Souleyrette, R., Hallmark, S., 2003. LIDAR vs. photogrammetry: comparison of cost and time, NCRST Report, URL: http://www.ncgia.ucsb.edu/ncrst/research/ncgia.html.

Steger, C., Glock, C., Eckstein, W., Mayer, H., Radig, B., 1995. Model-based road extraction from images, *Workshop on AEMOASI*, Basel, pp. 275-284.

Stilla, U., Hajdu, A., 1994. Map-aided structural analysis of aerial images, *IAPRS*, vol. 30, pp. 475-482.

Tao, V., 2002. Technology trends - digital photogrammetry: the future of spatial data collection, *GEO World*, URL: http://www.geoplace.com/gw/.

Tao, V., Hu, Y., 2001. A review of post-processing algorithms for airborne LIDAR data, *ASPRS Annual Conference*, 23-27 April, St. Louis, 12 p.

Tao, V., Hu, Y., 2002. Assessment of airborne lidar and imaging technology for pipeline mapping and safety applications, *Joint Conference on Integrating Remote Sensing at the Global, Regional and Local Scale*, 8-15 November, Denver, 11 p.

Tao, V., Wang, Y., Mercer, B., Zhang, Y., 2001. Automatic reconstruction of bald digital terrain models from digital surface models generated from an airborne SAR system, *International Symposium on Mobile Mapping Technology*, 3-5 January, Cairo, 11 p.

TerraSolid, 2001. *TerraScan User's Guide*, 118 p. URL: http://terrasolid.fi/.

Thompson, G.W., Maune, D.F., 2001. Floyd's Aftermath, *Point of Beginning*, URL: http://www.pobonline.com/.

Trinder, J.C., Wang, Y., 1998. Knowledge based road interpretation in aerial images, *IAPRS*, vol. 32, pp. 635-640.

Trinder, J.C., Wang, Y., Sowmya, A., Palhang, M., 1997. Artificial intelligence in 3-D feature extraction, *Workshop on AEMOASI*, 5-9 May, Ascona, pp. 257-266.

Umasuthan, U., Wallace, A.M., 1996. Outlier removal and discontinuity preserving smoothing of range data, *IEE Proceedings on Visual Imaging & Signal Processing*, 143(3): 191-200.

Vestri, C., Devernay, F., 2001. Using robust methods for automatic extraction of buildings, *CVPR*, vol. 1, 8-14 December, Kanai, Hawaii, pp. 133-138.

Veneziano, D., Hallmark, S., Souleyrette, R., 2002. Comparison of lidar and conventional mapping methods for highway corridor studies, URL: http://www.ctre.iastate.edu/.

Vigo, M., 1997. An improved incremental algorithm for constructing restricted Delaunay

triangulations, *Computers & Graphics*, 21(2): 215-223.

Vonderohe, A.P., 2003. LIDAR vs. photogrammetry: comparison of accuracy, NCRST Report, URL: http://www.ncgia.ucsb.edu/ncrst/research/lidaraccuracy/first.html.

Vosselman, G., 1999. Building reconstruction using planar faces in very high-density data, *IAPRS*, 8-10 September, Munich, vol. 32, 6 p.

Vosselman, G., 2000. Slope based filtering of laser altimeter data, *IAPRS*, 17-22 July, Amsterdam, vol. 33, part B3, pp. 935-942.

Vosselman, G., de Gunst, M., 1997. Updating road maps by contextual reasoning, *Workshop on AEMOASI*, 5-9 May, Ascona.

Vosselman, G., de Knecht, J., 1995. Road tracing by profile matching and Kalman filtering, *Workshop on AEMOASI*, Basel, pp. 265-274.

Vosselman, G., Dijkman, S., 2001. 3-D building model reconstruction from point clouds and ground plans, *IAPRS*, vol. 34, part 3/W4, pp. 37-44.

Wack, R., Wimmer, A., 2002. Digital terrain models from airborne laser scanner data – a grid based approach, *IAPRS*, 9-13 September, Graz, vol. 34, part 3A/B, pp. 293-296.

Wang, C., Hu, Y., Tao, V., 2003. Identification and risk modeling of airfield obstructions for aviation safety management, submitted to *Journal of Environmental Informatics*, 30 p.

Wang, Y., Mercer B., Tao, V., Sharma, J., Crawford, S., 2001. Automatic generation of bald earth digital elevation models from digital surface models created using airborne IfSAR, *ASPRS Annual Conference*, 23-27 April, St. Louis, 11 p.

Wang, Y., Trinder, J., 2000. Road network extraction by hierarchical grouping, *IAPRS*, vol. 33, pp. 943-949.

Wang, Z., Schenk, T., 2000. Building extraction and reconstruction from lidar data, *IAPRS*, 17-22 July, Amsterdam, vol. 33, part B3, pp. 958-964.

Wehr, A., Lohr, U., 1999. Airborne laser scanning - an introduction and overview, *ISPRS JPRS*, 54, 68-82.

Wiedemann, C., 2002. Improvement of road crossing extraction and external evaluation of the extraction results, *IAPRS*, 9-13 September, Graz, vol. 34, part 3A/B, 4 p.

Weidner, U., 1995. Building extraction from digital elevation models, 56 p. URL: http://www.ipb.uni-bonn.de/ipb/lit/abstracts95/weidner95.building.html.

Weidner, U., Forstner, W., 1995. Towards automatic building extraction from high resolution digital elevation models, *ISPRS JPRS*, 50, 38-49.

Wild, D., Krzystek, P., 1996. Automated breakline detection using an edge preserving filter, *IAPRS*, Vienna, vol. 31, pp. 946-952.

Williams, D., Shah, M., 1992. A fast algorithm for active contours and curvature estimation, *CVGIP: Image Understanding*, 55(1): 14-26.

Wu, H., Gong, J., Li, D., Shi, W., 2000. An algebraic algorithm for point inclusion query, *Computers & Graphics*, 24, 517-522.

Yoon, T., Kim, T., Park, W., Kim, T.G., 1999. Building segmentation using active contour model, *Joint ISPRS Workshop on Sensors and Mapping from Space*, 27-30 September, Hannover, 7 p.

Zhan, Q., Molenaar, M., Tempfli, K., 2002a. Building extraction from laser data by reasoning on image segments in elevation slices, *IAPRS*, vol. 34, pp. 305-308.

Zhan, Q., Molenaar, M., Tempfli, K., 2002b. Hierarchical image object-based structural analysis toward urban land use classification using high-resolution imagery and airborne lidar data, *International Symposium on Remote sensing of Urban Areas*, Istanbul, 8 p.

Zhang, S., 1993. *Numerical Algorithms in C Language* (in Chinese), Haiyang Press, Beijing, 600 p.

Zhang, C., Baltsavias, E., Gruen, A., 2001. Knowledge-based image analysis for 3D road reconstruction, *Asian Journal of Geoinformatics*, 1(4): 3-14.

Zhang, X., Burkhardt, H., 2000. Grouping edge points into line segments by sequential Hough transformation, *ICPR*, 3-8 September, Barcelona, pp. 3676-3679.

Zhang, T., Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns, *Communications of the ACM*, 27(3): 236-239.

Zinger, S., Nikolova, M., Roux, M., Maitre, H., 2002. 3D resampling for airborne laser data of urban areas, *IAPRS*, 9-13 September, Graz, vol. 34, part 3A/B, 6 p.

Zlotnick, A., Carnine, P.D., 1993. Finding road seeds in aerial images, *CVGIP: Image Understanding*, 57, 243-360.

APPENDIX A

AIRBORNE LIDAR TECHNOLOGY

Airborne lidar is an aircraft-mounted laser system designed to collect elevation data of the Earth's surface directly and digitally. The lidar instrument transmits laser light out to a target. Some of this light is reflected back to the instrument where it is analyzed. There are three basic types of lidar: the range finder, DIAL and Doppler lidar (NASA, 1999). Range finder lidar is used to measure the distance from the lidar instrument to a target. The time for the light to travel out and back to the lidar is used to determine the range to the target (Optech, 2003). "Range finder lidar uses the same principle as radar. It operates in the ultraviolet, visible or infrared region of the electromagnetic spectrum, whose wavelengths are much shorter than those used by conventional radar. Lidar may be continuous-wave (CW) or pulsed, focused or collimated. The CW lidar is used when the signal may be integrated over long time periods and when the target is nearby. They are convenient to use when measuring average properties of the path to the target. Focusing is mainly used with CW lidar to permit them to make a more sensitive measurement over a smaller span of ranges. Pulsed lidar uses much higher power levels during the laser pulse than can be maintained with a CW laser, producing higher signal-to-noise ratios for the collected radiation. Pulsed lidar is usually chosen for long-range remote sensing and when long signal integration is impractical" (NASA, 1999).

A.1 AIRBORNE LIDAR MAPPING

A typical airborne lidar system (see Figure A.1, URL: http://gis.esri.com/library/) is composed of a LRF, a POS, realized by the integration of a DGPS, an IMU and control units (Wehr and Lohr, 1999). Most lidar systems use a scanning mirror to generate a swath of light pulses. Swath width depends on the mirror's angle of oscillation and flight height, and the ground-point density depends on factors such as the aircraft speed, the oscillation rate of mirror, the laser pulse rate and the flying height (Baltsavias, 1999b; Sapeta, 2000). The laser measures the range to the ground surface or objects, and yields

the 3-D position when combined with the position and orientation of the sensor. The major characteristics of typical airborne lidar systems are summarized in Table A.1 (Baltsavias, 1999a; Fowler, 2001; Flood, 2001; Optech, 2003).

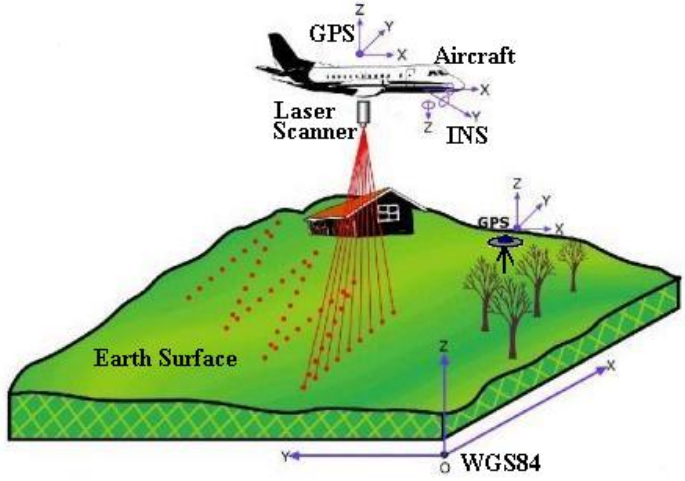


Figure A.1. A typical airborne lidar system

Table A.1. Major characteristics of typical lidar systems

Parameters	Min. & Max. values	Typical values
Laser wavelength (nm)	810 - 1550	1000-1200
Scan angle (degrees)	14 - 75	20-40
Pulse rate (kHz)	5 - 83	5-15
Scan rate (Hz)	20 - 630	25-40
Flying Height - *h* (m)	20 - 6100	200-300 (H) / 500-1000 (A)*
Swath width (m)	0.25 h - 1.5 *h*	0.35 h - 0.7 *h*
GPS frequency (Hz)	1 - 10	1-2
IMU frequency (Hz)	40 - 200	50
Beam divergence (mrad)	0.05 - 4	0.25-2
Footprint diameter (m)	0.05 - 2	0.25-1 (*h* = 1000 m)
Across-track spacing (m)	0.1 - 10	0.5-2
Along-track spacing (m)	0.06 - 10	0.3-1
Range accuracy (cm)	2 - 30	5-15
Elevation accuracy (cm)	10 - 60	15-20

Planimetric accuracy (m)	0.1 - 3	0.3-1

* H = helicopter, A = airplane.

A.1.1 Principle of Ranging Lidar

During a flight mission, an aircraft rotates on three different axes, commonly known as roll, pitch and yaw. The simplified relationship among the position and attitude of the laser scanner, the instantaneous mirror angle and the measured range is shown in Figure A.2 (Baltsavias, 1999b). The detailed and rigorous formulae can be found in El-Sheimy (1996), and PE&RS (2001), but are outside the research scope of this thesis.

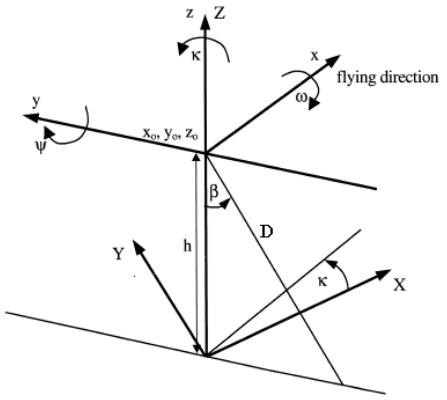


Figure A.2. The laser range, the scanner position and rotation angles

The LRF works a lot like ordinary radar, except that it sends out narrow pulses or beams of light rather than broad radio waves. It consists of two units: an opto-mechanical scanner and a laser ranging unit (Wehr and Lohr, 1999). The scanner comprises the laser transmitter and the electro-optical receiver. The LRF works as follows: the laser scanner generates an optical pulse; the pulse is reflected off an object and returns to the receiver; a high-speed counter measures the time of flight from the start pulse to the return pulse; and finally the time measurement is converted to the range *D* from the scanner to the object.

The integrated position and orientation system consists of a DGPS and an IMU (Wehr and Lohr, 1999; Mohamed and Price, 2002). The GPS is a constellation of 24 satellites that orbit the Earth and transmit signals. It allows one to calculate one's position on the

Earth to the accuracy of better than 10 cm in differential mode. The IMU consists of two trials of accelerators and gyroscopes, sensing incremental linear and angular velocities of a platform-fixed coordinate system. Inertial techniques are based on integrating the linear and angular velocities, which can then be used to calculate the platform position and three rotation angles (Mohamed, 2003). As shown in Figures A.1 and A.2, the scanner position (X_0, Y_0, Z_0) is provided by the GPS; and the IMU provides the platform rotation angles w (the angle around the flight direction), y (the angle around the across-track direction) and k (the angle around the perpendicular axis).

The control units manage the digital interfaces between LRF and POS (Wehr and Lohr, 1999). Geocoding of laser scanner measurements requires an exact synchronisation of both LRF and POS. The LRF measures only the spatial vector from the laser scanner to a ground point on the Earth's surface shot by a laser beam. Combining GPS and IMU information using advanced Kalman filtering techniques allows for the determination of position and attitude that are more accurate (Sapeta, 2000). The outcome is a complete set of exterior orientation data (i.e., X_0, Y_0, Z_0, w , y , k). To compute the 3-D position of a point, a recording-unit combines measured ranges, mirror scan angles, GPS positions and IMU orientation information at the epoch of each laser shot, and then performs a series of transformations to rotate and translate the laser range from a local aircraft coordinate system to WGS84 (Baltsavias, 1999b; Wehr and Lohr, 1999; Sapeta, 2000).

A.1.2 Lidar Data Acquisition

Today, there are many commercial lidar systems in the market delivering a variety of lidar mapping services (Baltsavias, 1999a; ALM, 2003). The entire process of airborne lidar mapping is highly automated, from flight planning, to data acquisition, to the production of DSMs. The parameters of flying height, swath angle, scanning rate, flight-strip side lap and aircraft velocity determine the point density, and these parameters are tailored to accommodate the project requirements (Baltsavias, 1999b; Sapeta, 2000). The accuracy of range data depends on the specific configuration of a lidar system. At present, claimed accuracies of commercial lidar systems are on the order of 15 cm vertically and 0.3~1 m horizontally (Flood, 2001; Fowler, 2001). Recent studies reveal

that vertical errors of less than 10 cm can be obtained at a low flying height (e.g., 400 m). In this case, DGPS and laser range noise dominate the error budget, and errors due to the IMU are secondary (Mohamed, 2003).

The need for accuracy validation and possible adjustment is due to the existence of residual systematic biases in the DGPS, IMU and ranging systems (Sapeta, 2000). "Adjustment techniques vary in sophistication, depending on the application and size of the project. Techniques range from simple vertical translations of mass points to remove vertical bias to complex block adjustments that involve ties among strips, ground controls and modeling systematic errors on a time-varying basis. Further, for high accuracy applications such as two-foot contouring, an adjustment to a network of GCPs may be required to guarantee a desired accuracy" (Sapeta, 2000).

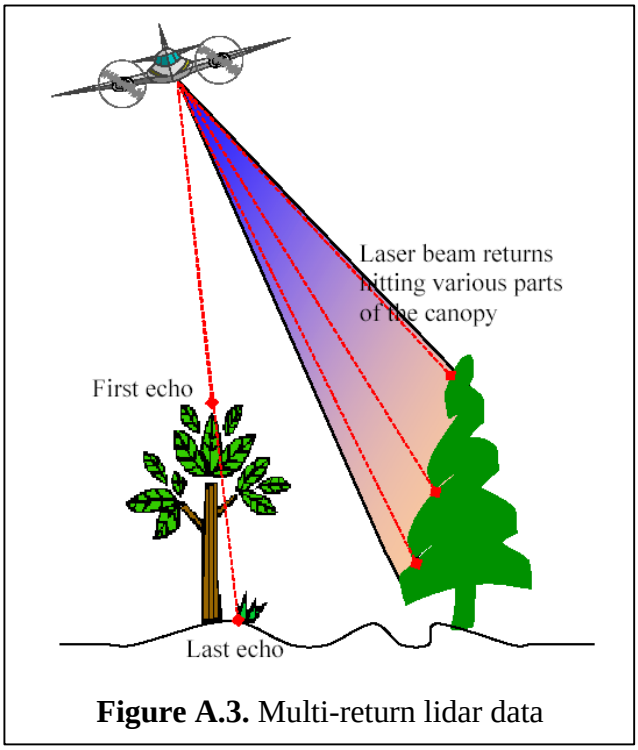


Figure A.3. Multi-return lidar data

In addition to range data, some lidar systems provide information on the intensity of the recorded signal, information for multiple parts of an object (see Figure A.3) within the laser footprint of a single pulse, and images taken by video cameras (Ackermanm, 1999; Mohamed et al., 2001; Optech, 2003; ALM, 2003). Figure 1.1 shows a sample lidar range

and intensity dataset collected at a village. Intensity capture of the returned pulse, through either waveform digitization or return pulse peak capture, is becoming increasingly common on commercial instruments (Flood, 2001; ALM, 2003). The intensity feature records how much energy is returned from the objects. The reflectivity rates of typical materials are listed in Table A.2 (Song et al., 2002). The multiple returns are especially useful in discriminating vegetation. When a beam has distended to a foot in diameter by the time it reaches objects, a unit may record the reflection of the canopy of a tree or the roof of a house and the reflection from the ground as two different elevations from the single pulse (Sapeta, 2000; Optech, 2003). As algorithm development continues, intensity and multi-return information will enable users to differentiate between different object classes and eventually help to automate the extraction of more features (Tao and Hu, 2001; Song et al., 2002; ALM, 2003).

Table A.2. Reflectivity of infrared laser

Material	Reflectivity (%)
White paper	Up to 100
Snow	80-90
Beer foam	88
Limestone	Up to 75
Deciduous trees	60
Toilet paper	60
Dry sand	57
Wet sand	41
Coniferous trees	30
Concrete	24
Asphalt with pebbles	17
Black neoprene	5

A.1.3 Lidar vs. Photogrammetry

The capabilities and limitations of lidar are often compared with the passive photogrammetry. Lidar has both advantages and disadvantages over photogrammetry.

- *Advantages of lidar over photogrammetry*:
  - Lidar is less expensive of producing the same data than photogrammetry especially for linear corridors (Sapeta, 2000; Fowler, 2001; Burst, 2002). Obtaining field measured terrain models using ground-based techniques may be more accurate, but are also more costly for a wide area (ALM, 2003).
  - Lidar can acquire accurate elevation data in decimetre while flying from a high altitude (Optech, 2003). "The accuracy of photogrammetric data is inversely proportional to the flying height, whereas lidar accuracy degrades less significantly with increased flying heights" (Thompson and Maune, 2001).
  - Lidar needs only a single laser pulse to measure the ground elevation, whereas photogrammetry requires two different lines of sight to both "see" the same points on the ground from two different perspectives (Thompson and Maune, 2001; Burst, 2002; Optech, 2003). Therefore, lidar has far fewer areas where the terrain is obscured by trees that block the lines of sight.
  - Lidar can collect both first and last returns with the capability to penetrate between foliage (Sapeta, 2000; Optech, 2003). Photogrammetry can generate high-density elevation points also, but only by expensive manual compilation of individual points, or by automated image correlation, but normally of treetops instead of the ground below (Thompson and Maune, 2001).
  - Lidar systems are less sensitive to environmental conditions such as weather, sun angle, leaf on/off condition (Fowler, 2001). Lidar can also work at night without the degradation in performance (Optech, 2003).

- *Disadvantages of lidar over photogrammetry:*
  - "Lidar is a relatively new technology, and standard procedures have not yet been developed to yield data with predictable accuracy comparable to that from photogrammetry" (Thompson and Maune, 2001; Vonderohe, 2003).
  - Infrared laser pulses are often absorbed by water except the use of blue-green lights (Optech, 2003). Normally, digital orthophotos are used to determine the limits of water boundaries (Thompson and Maune, 2001).

- Lidar data are not well suited to determine breaklines (Fowler, 2001; Burst, 2002). Breaklines compiled photogrammetrically are often used to augment the lidar data as needed for improving terrain modeling (Veneziano et al., 2002).
- Contours generated automatically from lidar data are more jagged, whereas contour lines produced manually by photogrammetric compilation are normally smooth to reflect the actual representation of the terrain and to depict the flow of water downstream (Thompson and Maune, 2001; Burst, 2002).

A.2 LIDAR MAPPING APPLICATIONS

The high point sampling densities, fast turn-around time and lower cost are among the most attractive characteristics of lidar for many mapping applications (Tao and Hu, 2001). There are many applications in favour of lidar data as described below:

- *Topography*. The DTM generation is the most direct application of lidar data (Fowler, 2001). It provides a cost-effective source of high-density elevation data for calculating terrain parameters and DTM derivatives in engineering mapping. These terrain parameters can be used to locate steep slopes, which could endanger nearby infrastructures (Tao and Hu, 2002). It also allows large area topographic surveys to be completed significantly faster and at a reduced cost compared to traditional mapping methods (Souleyrette and Hallmark, 2003). There are other cases where terrain structures are discernible and can be derived from the geometric information contained in high-density lidar data (Kraus and Pfeifer, 1998; Kraus and Rieger, 2000; Fraser and Jonas, 2001). For instance, breaklines of the terrain can be extracted to some extent (Wild and Krzystek, 1996).
- *Urban modeling.* Since lidar systems provide dense measurements, the extraction of 3-D features with sharp discontinuities, especially buildings, is easier than in surface models obtained by stereo matching, or faster than manual compilation (Ackermanm, 1999; Tao and Hu, 2001). Different perspective views can be generated from different positions and directions for a variety of applications including landscape analysis, wireless communication, urban planning, pilot

simulation, microclimate modeling, disaster response and propagation of noise and pollutants (ALM, 2003; Optech, 2003).

- *Right-of-way.* A major market is the transportation industry. Since lidar has a narrower swath in comparison to optical sensors, it is more cost-effective in capturing information for linear features (Veneziano, 2002; ALM, 2003). Lidar provides rapid range data collection of long, linear objects such as roads, railway tracks, pipelines, waterways, coastal zone and power lines (Hill et al., 2000; Burst, 2002; Optech, 2003). The power line corridor mapping allows for proper modeling of conductor catenary curves, encroachment and accurate determination of tower locations (TerraSolid, 2001).
- *Forestry.* Accurate information on the terrain and topography beneath the tree canopy is extremely important to both the forestry industry and natural resource management (ALM, 2003). Tree heights, crown diameter and density information may be produced from multi-return lidar data, while they are difficult to be collected using traditional methods (Optech, 2003). The derivation of other important parameters like biomass estimation, tree type etc. is also possible.
- *Change detection.* Lidar can be used to efficiently locate areas of change, which are invaluable information for subsequent map updating. Lidar data taken during the first mapping phase are compared to later datasets, and areas of change can be quickly detected by simply computing and thresholding a difference image (Murakami et al., 1999; Sapeta, 2000). This is important when terrain surfaces change during natural disasters such as landslide.

Since lidar is a relatively new technology, applications are still being identified as users start to work with the data. Industrial and scientific advances in lidar systems and data processing techniques are opening new technological opportunities to develop an increased capability to accomplish many geospatial and engineering applications.

APPENDIX B

BASIC ALGORITHMS AND CONCEPTS

This appendix describes algorithms and concepts for image processing, computational geometry, numerical analysis and quality measures. I developed two new algorithms, i.e., the synthesized edge detection and the incremental connected component labelling.

B.1 IMAGE PROCESSING ALGORITHMS

B.1.1 Introduction

The image can be represented by a continuous function $I(x, y)$, where x and y denote coordinates in a 2-D Cartesian system $\Re^2$ as shown in Figure B.1a. The digital number (DN) I at a point (x, y) is proportional to certain property (e.g., the light intensity and terrain elevation) of the scene at that point (Gonzalez and Woods, 2002). A digital image with M rows and N columns is an image that has been discretized in spatial coordinates and the DN, and maybe denoted as $I(i, j)$, where i ($0 \le i < M$) and j ($0 \le j < N$) are zero-based integer row and column indices as shown in Figure B.1b. In this thesis, the pixel coordinate (M-1, 0) corresponds to the point (0.5, 0.5) in spatial coordinate system.

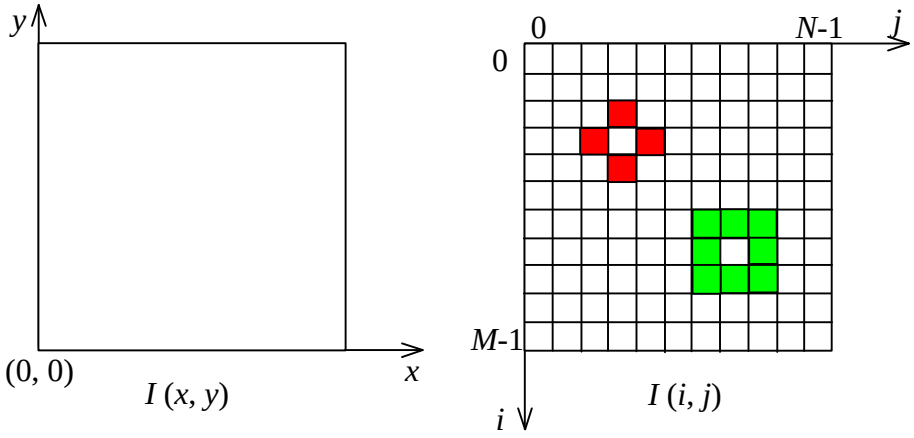
Many algorithms have to formulate an adjacency criterion for pixel connectivity. For a pixel p with the coordinates (i, j), the set of pixels given by (Gonzalez and Woods, 2002)

$$N_4(p) = \{(i, j+1), (i-1, j), (i, j-1), (i+1, j)\} \tag{B.1a}$$

is called its 4-neightborhood, that is the red cells for $p(3, 3)$ in Figure B.1b. The 8-neighborhood (see green cells for $p(7, 7)$ in Figure B.1b) is defined as

$$\begin{aligned} N_8(p) = \{&(i, j+1), (i-1, j-1), (i-1, j), (i-1, j-1), \\ &(i, j-1), (i+1, j+1), (i+1, j), (i+1, j+1)\} \end{aligned} \tag{B.1b}$$

Two pixels *p* and *q* are called 4-connected if *q* is from $N_4(p)$, or 8-connected if *q* is from $N_8(p)$. In a binary image, it is reasonable to define the connectivity as 4-connected for (0-valued) background regions and 8-connected for (1-valued) object regions.



(a) Spatial coordinate system (b) Pixel coordinate system

Figure B.1. Image coordinate systems

B.1.2 Image Pyramid

The image pyramid is a stack of images of decreasing resolutions, each representing pattern information at a different scale. The basic idea is to gradually reduce the spatial resolution of the image and decrease the amount of information by following filters.

- "*Linear filters* convolve an image with linear filter kernels, and thus output the pixel value as a linear combination (e.g., a weighted average) of the values of the pixels in certain neighborhood" (Pitas and Venetsanopoulos, 1990).
- *Non-linear filters* calculate a non-linear function of the pixel values using order statistic, homomorphic, or polynomial filters (Pitas and Venetsanopoulos, 1990).
- *Morphological filters* are a special type of non-linear filters applied to the image, and then the filtered image is sub-sampled to the reduced size (Pitas and Venetsanopoulos, 1990). They are often closely related to order statistic filters.
- *Wavelet filters* produce a hierarchical decomposition of functions, here images. Wavelets provide means of frequency and space analysis. A function is described

by means of a low-resolution function plus a series of details from low to high resolution based on wavelet coefficients (Mallat, 1989).

B.1.3 Morphological Filters

Morphological operations tend to simplify, enhance, extract or describe images using set theory (Li, 1999). The set represents a binary or gray image. Morphological filters extract information about the geometrical structure of an image object by filtering it with a structuring element, which is of simpler shape than the original object. Information about size, spatial distribution, shape, connectivity, convexity, smoothness, and orientation can be obtained by transforming the image object using different structuring elements (Pitas, 2000). Symmetrical and circular structuring elements play a central role in mathematical morphology in the continuous plane, because they provide an isotropic treatment of the image. A large number of task-oriented procedures can be built with morphological operators. These tasks include noise suppression, boundary detection, skeletonization, thicken, feeler pruning, the convex hull and region filling techniques (Gonzalez and Woods, 2002). For digital images, rectangular windows are often used because they are easier and faster to implement. Basic morphological operators include erosion, dilation, opening and closing, which are defined below (Pitas and Venetsanopoulos, 1990).

The dilation of a set of points A by a structuring element S is defined by Equation B.2a, where S_t is a translation operation of S by a vector $t \in \Re^2$, i.e., $S_t = \{s + t \mid \forall s \in S\}$.

$$A \oplus S = \{t \mid S_t \cap A \neq \Phi\} \tag{B.2a}$$

The erosion is the opposite of dilation, and is defined as

$$A \Theta S = \{t \mid S_t \subset A\} \tag{B.2b}$$

The opening, denoted as $A \circ S$, and closing, $A \bullet S$, are given by Equations B.2c and B.2d, respectively, where $\breve{S}$ is the reflection operation defined as $\breve{S} = \{-s \mid \forall s \in S\}$.

$$A \circ S = (A \Theta S) \oplus \breve{S} \quad \text{(B.2c)}$$

$$A \bullet S = (A \oplus S) \Theta \breve{S} \quad \text{(B.2d)}$$

B.1.4 Median and Multi-stage Median Filters

Noise usually appears in the high frequencies of the image spectrum. Therefore, a low-pass digital filter may be used for noise removal. Such a filter is the moving average filter. However, linear low-pass filters tend to smear image details (e.g., lines, corners), whose power is in the high frequencies as well. Furthermore, they tend to blur the image edge for similar reasons, thus degrading the image quality. One way to keep the edges intact is to use a median filter, in which the median value over a small neighborhood is used to replace the central pixel. Median filters have low-pass characteristics to remove additive white noise (Pitas and Venetsanopoulos, 1990). They are very efficient in the removal of noise having a long-tailed (e.g., Laplacian) distribution. The robustness properties of the median filter make it very suitable for edge-preserving filtering.

In the theoretical analysis of median filters, it is assumed that the image consists of constant neighborhoods and edges (Pitas and Venetsanopoulos, 1990). In reality, the images have fine details, for example, lines and sharp corners, which are very valuable for human perception. These details are usually destroyed by medians having relatively large windows. It is the ordering process that destroys the structural and spatial neighborhood information. The multi-stage median filter can preserve details in horizontal, vertical and diagonal directions, because they use sub-filters that are sensitive to these directions (Arce and Foster, 1989).

B.1.5 Adaptive Smoothing

Smoothing is needed to filter out local fluctuations in the data due to measurement noise to obtain reasonable differential geometric quantities from a digital surface. The linear smoothing often makes sharp discontinuities in range and orientation blurred. Research has explored the idea that the unwanted effects of smoothing can be eliminated, or at least significantly attenuated, by using adaptive smoothing and derivative estimation operators. That is, window operators may change shape and size depending on local data variations

(Besl, 1988). Several techniques, such as adaptive smoothing or anisotropic diffusion deal with the task of local smoothing. Unfortunately, these types of iterative techniques have one main drawback, i.e., the determination of the threshold on the gradient. There is no way to control it easily and researchers often fall into a trial-and-error procedure.

Saint-Marc et al. (1991) developed a method to smooth a signal while preserving discontinuities. This is achieved by repeatedly convolving the signal with a 3 x 3 averaging mask weighted by a measure of the signal continuity at each point. Edge detection can be performed after a few iterations, and features extracted from the smoothed signal are correctly localized (hence, no tracking is needed). This last property allows the derivation of a scale-space representation of a signal using the adaptive smoothing parameter *k* as the scale dimension. A scheme to preserve higher-order discontinuities and results on range images is proposed. The algorithm iteratively performs weighted averaging defined by (Saint-Marc et al., 1991)

$$I^{(t+1)}(i,j)=\frac{\sum_{u=-1}^{+1}\sum_{v=-1}^{+1}I^{(t)}(i+u,j+v)w^{(t)}(i+u,j+v)}{\sum_{u=-1}^{+1}\sum_{v=-1}^{+1}w^{(t)}(i+u,j+v)} \tag{B.3}$$

with

$$w(i,j)=\exp\left(-\frac{I_x^2(i,j)+I_y^2(i,j)}{2k^2}\right)$$

$$I_x(i,j)=[I(i+1,j)-I(i-1,j)]/2$$

$$I_y(i,j)=[I(i,j+1)-I(i,j-1)]/2$$

where *t* is the iteration number; *k* is the smoothing parameter; *I* is the image; *w* is the weighting factor. Only discontinuities larger than *k* will be enhanced during iterations.

B.1.6 Segmentation and Thresholding

The purpose of segmentation is to separate an image into regions corresponding to objects, that is, to divide the image into homogeneous regions, which have common

properties (Geiger and Yuille, 1991). Image regions are expected to have homogeneous characteristics (e.g. grey level, or colour), which may indicate that they belong to the same object or facet of an object. Segmentation by thresholding has been refined into a set of mature procedures and the outstanding problem is how to devise an automatic procedure for determining optimal thresholds.

Thresholding is useful in discriminating objects from the background in many classes of a scene. Most thresholding techniques indirectly utilize the shape information of an image histogram. The ideal case is a bimodal shape of histogram because grey level at the valley can be directly selected as threshold value for segmentation, but such bimodality histogram is usually unavailable in real applications (Parker, 1993). In general, an image grey level can be divided into several sub-ranges to perform thresholding but these ranges usually overlap one with another and make thresholding difficult. Glasbey (1993) described an iterative selection method to calculate thresholds. The initial threshold T is the average gray level of the whole image. The average levels of both the background (T_b) and the objects (T_o) defined by T are calculated. The threshold T is recalculated to be half way between these two means, that is the integer part of $(T_b + T_o) / 2$. Then T_b and T_o are recalculated, and a new value of T is obtained. This is repeated until a repeat of the same value of T on two consecutive iterations (Glasbey, 1993). Papamarkos and Gatos (1994) proposed an approach for multilevel threshold selection.

B.1.7 Connected Component Analysis

For binary image, individual connected components of each region can be identified and labeled. Connected component labeling counts the objects that exist in a binary image. It assigns a unique number to each block of object pixels. Labelling algorithms can be divided into two large categories: local neighborhood algorithms and divide-and-conquer algorithms (Pitas, 2000). “The algorithms belonging to the first class perform iterative local operations. They use the grassfire concept implemented in a recursive manner. The image is scanned in a row-wise manner until the first object pixel is hit. The connected component is set at the first pixel that propagates to all object pixels belonging to a neighborhood of the current pixel. The operation is continued recursively until all object

pixels are burnt and the fire is extinguished. At the end of this operation, all pixels belonging to this object have the same value" (Pitas, 2000). If the object is very large, it is likely to result in a stack overflow during recursive labeling procedure. To avoid stack overflow, I develop an incremental labeling algorithm by integrating both local neighborhood and divide-and-conquer operations. The incremental method labels only a limited number (e.g., 256) of pixels at a time; then those connected components having different labeling values are merged using a same value.

B.1.8 Thinning

Image thinning methods can be divided into two categories based on the type of image they are designed to thin: binary or gray scale. Since binary image thinning methods cannot be directly applied to gray images, gray images are often thresholded to create a binary counterpart to which a binary method may be applied. Binary image thinning can be defined heuristically as a set of erosions of the outermost layers of a binary shape, until a connected unit-width set of lines is obtained. Zhang & Suen (1984) proposed a fast parallel thinning algorithm that requires two successive iterative passes. At Step 1, a logical rule is applied locally in a 3 x 3 neighborhood to flag border pixels that can be deleted. These pixels are only flagged until the entire image is scanned. Deletion of all flagged pixels is performed afterward. At Step 2, another logical rule is applied locally in 3 x 3 windows to flag border pixels for deletion. After the entire image has been scanned, the flagged pixels are again deleted. This procedure is repeated until no more pixel deleting can be performed. End points and pixel connectivity are preserved.

B.1.9 Synthesized Edge Detection

Edges maybe the most important low-level feature. Edge operators usually detect local maxima or zero crossings after applying first- or second-order derivative filters. Optimal edge operators focusing the step edge model and the additive white noise model have been developed. Canny (1986) proposed three criteria including sensibility, localization and immunity for evaluating the performance of edge operators. The sensibility criterion expresses the fact that important edges should not be missed, and that there should be no spurious responses. The localization criterion says that the distance between the actual

and located position of the edge should be minimal. The immunity criterion minimizes multiple responses to a single edge. Canny operator is the result of solving an optimisation problem with above constraints using variation principle.

In general, we have to use multiple Gaussian operators with different standard deviations because the signal-to-noise ratio is likely different for edges in the image. Canny (1986) proposed to aggregate the final information about edges at multiple scales using the feature synthesis approach. The response of a larger operator can be predicated from responses of smaller operators. If the response of the larger operator is significantly different from the predicated value, then it is discriminated as new edge points. In most cases, most edges are detected by small operators, and large operators detect edges at shadows or between texture regions. The feature synthesis approach usually produces a map of dense edge points and the edge points detected from smaller operators also present shifts in large scale. Bergholm (1987) proposed an edge focusing method to merge edges in coarse-to-fine manner, but distortions happens for some blurred edges at small scales. Both methods cannot determine which scale is best in different situations.

I modify the feature synthesis approach to detect edges by directly synthesizing multi-scale information. This synthesized edge detection uses the following two steps to compute the magnitude of edge strength and to estimate local edge normal directions. Let the 2-D Gaussian function with a standard deviation s be (Canny, 1986; Zhang, 1993)

$$G(x,y)=\frac{1}{2ps^{2}}\exp(-\frac{x^{2}+y^{2}}{2s^{2}}) \qquad \text{(B.4)}$$

- Compute edge strengths in x and y directions using 1-D Gaussian operators for n different scales s_i (i=1…n). The sizes of Gaussian masks are $2\sqrt{2}s_i+1$.

$$E_{xi}=G_{xi}*I(x,y)$$
$$E_{yi}=G_{yi}*I(x,y)$$

with

$$G_{xi} = -kx\exp(-x^2/2s_i^2)\exp(-y^2/2s_i^2) = -h_1(x)h_2(y)$$
$$G_{yi} = -ky\exp(-y^2/2s_i^2)\exp(-x^2/2s_i^2) = -h_1(y)h_2(x)$$
$$h_1(t) = \sqrt{k}t\exp(-t^2/2s_i^2) = t\cdot h_2(t)$$
$$h_2(t) = \sqrt{k}\exp(-t^2/2s_i^2)$$
$$k = 1/(2ps^4)$$

- Accumulate edge strengths at multiple scales to obtain the synthesized edge strength E and the gradient or normal direction b at point (x, y).

$$E(x,y) = \sqrt{E_x^2(x,y) + E_y^2(x,y)} \qquad \text{(B.5a)}$$
$$b(x,y) = \arctan\left(E_y(x,y)/E_x(x,y)\right) \qquad \text{(B.5b)}$$

with

$$E_x = \sum_{i=1}^{n} w_i E_{xi}, \quad E_y = \sum_{i=1}^{n} w_i E_{yi}, \quad w_i = 1/n$$

The experimental results show that the large and small operators give similar responses at places where strong discontinuities occurs; while large operators play the dominated role at relatively smooth areas. Most edges are detected by large operators in my method, and small operators mainly make the edge location more accurate and find some details. The number of edge points is also reduced. This is contrary to the feature synthesis approach. The synthesized edges are composed of both contours in large scale and details in small scale. Moreover, the responses from small operators can rectify the location shift resulted due to large operators, and the large operators can depress redundant details. Therefore, the edge distortion and location shift when applying the coarse-to-fine merging (Bergholm, 1987) and fine-to-coarse prediction (Canny, 1986; Lacroix, 1990) are both

conquered. In addition, more details can be obtained if larger weights are assigned to small operators when synthesizing edge strengths.

B.1.10 Chain Coding

Chain coding is a popular and efficient method permitting lossless contour or boundary representation. A curve is represented by the absolute image coordinates of its starting pixel followed by an ordered chain of vectors connecting between successive boundary pixels with a limited set of possible directions. The directional vectors are encoded as the digits from 0 to 7 or 0 to 3, assigned to the 8- or 4-neighboring pixels in counter-clockwise as defined in Equation B.1a or B.1b. Each digit is coded according to the relative position of a pixel to its predecessor. In an image with size M by N, a chain of length n needs bits defined by $n_{bits} = \log(M) + \log(N) + (n\text{-}1)\log(k)$, where k represents 4- or 8- neighboring. The required bits can be reduced by expressing the chain by the derivative chain code, which is based on the fact that, starting from a given pixel, there are only a limited number of possibilities to continue a chain. Then various characteristics, such as area, length, size, centroid, shape regularity, etc. can be calculated for a closed chain, which may represent a building boundary (Parker, 1993).

The derivative chain code method has many applications. Kaneko and Okudaira (1985) described a chain segmentation algorithm. The encoding scheme takes advantage of the property that a curve with smooth curvature can be divided into segments, each of which is represented by a sequence of two adjacent-direction derivative codes. In each segment, the derivative code is only composed of two symbols, reducing the per segment entropy. This is possible because of the hypothesis of smooth curvature within one segment. Therefore, the coding efficiency of the scheme becomes higher as the segments become longer. A similar approach is used in Baruch and Loew (1988), where the contour lines are segmented into segments with constant curvature, i.e. segments that are either straight lines or circular arcs.

B.1.11 Active Contour Models

A chain encoded from edge points may be open at both ends, and have to be closed to represent the object boundary. The active contour model, also known as snakes, can model a closed contour to the boundary of an object in an image. The contour may be

initialized using the encoded object edge points with a distance between adjacent vertices of a few pixels. The snake model is an energy-minimizing contour that deforms under the influence of internal forces, image forces, and external constraint forces (Kass et al., 1987). A snake is represented as a parametric curve, i.e., $v(s) = (x(s), y(s))$, where s runs from 0 to 1 over the perimeter of the snake.

Snake is controlled by minimizing a function, which converts high-level contour information like curvature and discontinuities and low-level image information like edge gradients and terminations into energies. Williams and Shah (1992) developed a greedy algorithm, which allows a contour with controlled first and second order continuity to converge in areas with high image energy. The energy function is defined as

$$E_{snake} = \int_0^1 \left(a(s)E_{continuity}(v(s)) + b(s)E_{curvature}(v(s)) + g(s)E_{image}(v(s)) \right) ds \qquad \text{(B.6)}$$

where a, b, g are parameters used to control the sensitivity between three energy terms. The continuity force $E_{continuity} = \bar{d} - | v_i - v_{i-1} |$ requires even spacing of vertices, where $\bar{d}$ is the average distance between vertices. The curvature force $E_{curvature} = | v_{i-1} - 2v_i + v_{i+1} |^2$ gives a reasonable curvature estimate. The shift towards areas where the gradient in the image is large is encouraged by the image force $E_{image} = (G_{\min} - G_{mag}) / (G_{\max} - G_{\min})$, where G_{max} and G_{min} are the maximum and minimum gradient in a neighborhood, and G_{mag} is the gradient of the vertex. The energy function is calculated at each vertex v_i and its neighbours. The new position of v_i is the location achieving the smallest energy.

I adaptively decrease the values of the curvature coefficient b when a right angle at some vertex is present. This is because a right angle indicates a corner in the contour at that vertex. Self-intersection of the contour is prohibited by increasing the curvature coefficient value to a very large value. This process closes the labelled open edge chains but with low modeling quality (Hu and Tao, 2002). By examining all the contours, the size of the largest buildings can be determined.

B.1.12 Hough Transformation

Hough transform (HT) allows for detecting the best fitting line/plane from a set of 2-D/3-D points, and is very robust to noise due to occlusions and false positives. The 2-D HT is applied to every 2-D point (x, y) for line detection as given by (Duda and Hart, 1972)

$$r = x\cos q + y\sin q \qquad \text{(B.7)}$$

where q is the angle of the line's normal with the x-axis; r is the algebraic distance from the origin to the line. Usually the angle is restricted to the interval of [0^o, 180^o), thus the orientation of a line is unique. Usually there are many types of edges in an image, and edges can be divided into many pieces of line segments. After pixels are transformed to Hough space, the computed accumulator may present several peaks. Lines can be found by detecting these peaks in the parameter domain. This is equivalent to the classification of the points into different groups of line points (Zhang and Burkhardt, 2000).

B.2 GEOMETRY AND NUMERICAL ALGORITHMS

B.2.1 Surface-Curvature-Sign Labeling

Differential geometry states that local surface shape is uniquely determined by the first and second fundamental forms (Besl, 1988). The mean and Guassian curvatures combine these first and second fundamental forms in two different ways to obtain scalar surface features that are invariant to translation, rotation, and change in parameterisation. Therefore, visible surfaces in range images have the same mean and Guassian curvatures from any viewpoint under orthographic projection. Eight fundamental viewpoint independent surface types can be characterised using only the sign of the mean curvature and Guassian curvature, including peak, pit, ridge, valley, saddle ridge, saddle valley, flat and minimal (Besl and Jain, 1988).

The mean curvature uniquely determines the shape of graph surfaces if a boundary curve is also specified while Guassian curvature uniquely determines the shape of convex surfaces and convex regions of non-convex surfaces. The mean and Guassian curvatures

can be computed directly from a range image using window operators that yield least squares estimates of first and second order partial derivatives (Besl and Jain, 1988). The key point is that every pixel can be given a surface type label based on the values of the pixels in a small neighborhood of that pixel.

B.2.2 Polygon

A polygon is a two-dimensional surface stored as a sequence of points defining its exterior bounding ring and zero or more interior rings. Polygons by definition are always simple, that is, not self-intersected. The exterior and any interior rings define the boundary of a polygon, and the space enclosed between the rings defines the polygons interior. The rings of a polygon can intersect at a tangent point but never cross. A number of different methods can be applied to compute the area of a polygon with vertex coordinates. The origin of the coordinate system is often translated to the centre of gravity of the dataset, to improve the numerical condition by avoiding subtraction and multiplication of large numbers (O'Rourke, 1998; Parker, 1993).

The 2-D point-in-polygon test is to determine if a point falls inside a given polygon. The popular algorithm uses the parity rule, in which a line is drawn from the point to some point that is guaranteed to lie outside the polygon. If the line crosses the polygon edges an odd number of times, the point is inside the polygon, otherwise it is outside (O'Rourke, 1998). The second algorithm is based on the winding number of the point, which is the number of revolutions made around that point while travelling once along the polygon. The point is inside the polygon if the winding number is nonzero (Wu et al., 2001).

B.2.3 Douglas-Peucker Curve Simplification

The most widely used high-quality curve simplification algorithm is probably the heuristic method called Douglas and Peucker algorithm. "At each step, the algorithm attempts to approximate a sequence of points by a line segment from the first point to the last point. The point farthest from this line segment is found, and if the distance is below a predefined threshold, the approximation is accepted, otherwise the algorithm is recursively applied to the two sub-sequences before and after the chosen point. This

algorithm has generally produced better approximations when compared with many other simplification algorithms" (Heckbert and Garland, 1997). The worst-case complexity is $n \cdot \log(n)$ using the geometric structure based on convex hulls, where *n* is the number of points (Hershberger and Snoeyink, 1992). A potential problem is that the simplification may cause a simple polygon to become self-intersecting.

B.2.4 Triangulated Irregular Networks

The most popular triangulation method is the Delaunay triangulation (DT), which does not use height values and is a 2-D method. DT finds the triangulation that maximizes the minimum angle of all triangles, among all triangulated irregular networks (TINs) of a given point set (De Berg et al., 2000). This helps to minimize the occurrence of very thin triangles. DT has many nice properties that make it very popular in surface modeling. In a DT, the circum circle of each triangle contains no vertices in its interior. DTs can be computed using the incremental insertion algorithm, the divide-and-conquer algorithm, and the plane-sweep algorithm (O'Rourke, 1998). Vigo (1997) developed an algorithm for constructing the constrained DT of a generalized planar graph. The algorithm works incrementally, and allows for the addition of a point to the constrained DT as well as the addition of an edge that constraints it.

B.2.5 Linear and Non-linear Regressions

Let the following linear system describe the parametric model (Shen et al., 1992):

$$l = Ax + v \tag{B.8}$$

where *l* is a vector of observations; *A* is the design matrix; *x* is a vector of unknown parameters; *v* is the error vector that are assumed to be independently and identically distributed. The solution to the unknowns can be obtained by a conventional least-squares (LSs) method (Shen et al., 1992):

$$x = (A^T C_l^{-1} A)^{-1} A^T C_l^{-1} l \tag{B.9}$$

where C_l expresses the a priori variance-covariance matrix of the observation vector. For the non-linear system, a Taylor expansion is often used to perform a linearization on those observation equations toward those unknowns. Then the parameter corrections are computed by applying the linear regression. The a priori values of the unknowns are corrected iteratively until certain termination condition is satisfied.

B.3 QUALITY MEASURES

Several quality measures are defined to assess the performance of a feature extraction algorithm based on the following quantities (Fischler and Heller, 1998; Wiedemann, 2002):

- *True positives*: primitives in both the extracted results and the reference data.
- *False positives*: primitives in the extracted results but not in the reference data.
- *False negatives*: primitives in the reference data but not in the extracted results.

The quality measures, i.e., completeness and correctness, are (Fischler and Heller, 1998; Wiedemann, 2002):

- *Completeness measure*: the percentage of the reference primitives that are extracted by the algorithm, i.e.,

$$QM_{completeness} = \frac{number\ of\ matched\ reference\ primitives}{number\ of\ all\ reference\ primitives} \times 100 = \frac{TP}{TP + FN} \times 100$$

- *Correctness measure*: the percentage of correctly extracted primitives with respect to all extracted ones, i.e.,

$$QM_{correctness} = \frac{number\ of\ matched\ extracted\ primitives}{number\ of\ all\ extracted\ primitives} \times 100 = \frac{TP}{TP + FP} \times 100$$

APPENDIX C

LIDAR EXPERT AND ITS APPLICATIONS

This appendix first describes Lidar Expert, a software package that was developed by me for the thesis work. Then three mapping applications using features extracted by Lidar Expert are presented.

C.1 LIDAR EXPERT

Lidar Expert has implemented most algorithms presented in previous chapters for extracting cartographic features from lidar range and intensity data. It now includes automated tools for pre-processing raw lidar data, generating bald earth terrain models, detecting road ribbons, reconstructing grid road networks, detecting building footprints, creating building models, mapping forested areas, computing pipeline risks, perform image processing, exporting extracted results, converting between different data formats and conducting other analyses. Users can configure the system by setting values for the parameters used by above tools.

The pre-processing tools perform blunder detection, point density determination, points thinning, vector to raster conversion etc. The DTM generation tool utilizes a hierarchical terrain recovery algorithm to intelligently discriminate between terrain and non-terrain points and produces bare Earth models with a high accuracy for scenes with typical terrain and coverage types. The road detection tool outputs road ribbons by classification lidar intensity and height data. Especially in built-up areas, it can reconstruct grid road networks with cartographical quality and calculate their attributes, including height of road crossings, and width, length and slope of road segments. The building extraction tool has two sub-procedures, i.e., building detection and building reconstruction. The detection step locates building footprints from lidar height data by segmentation; then computes orthogonally rectified polygonal boundaries. The reconstruction step creates prismatic models for flat roof buildings and polyhedral models for non-flat roof buildings.

This process also calculates the attributes of building models, including orientation, area, perimeter, size, average/maximum height, footprint type, roof type and so on.

Lidar Expert can import single- and bi-return lidar data in ASCII and binary formats including TIFF with world files, ArcView grid, PNG, PPM and PIX. It exports results in image formats such as TIFF, PNG and PIX for grid data, in vectors formats such as ESRI Shapefile for 3-D building models and road networks, and in ASCII format for TIN representation. However, Lidar Expert itself does not display the results at present. To visualize the extracted results, users have to use other commercial systems, such as ArcGIS, PCI Geomatica, or ERDAS IMAGINE, to combine exported files.

The tools included in Lidar Expert have been tested using many lidar datasets collected at areas of varying scene complexities. The point densities change from 0.43 to 2.8 points/m^2. The results show that in most areas the lidar-derived DTMs retain most terrain details and remove non-terrain objects reliably; the road ribbons and grid road networks are sketched well in urban areas; and the extracted building footprints have high positioning accuracy equivalent to ground observations.

C.2 3-D CITY MODELS

C.2.1 Introduction

The production and use of 3-D city models are important tasks for a variety of geospatial applications, including urban planning, traffic control, environmental studies, marketing, telecommunication, battlefield simulation, and propagation of pollutants (Haala and Anders, 1997; Brenner, 2000; Tao and Hu, 2001). With 3-D city models, different perspective views can be generated from different positions and directions for landscape analysis. Telecommunication companies can optimize the distribution of base station and antenna sites with the line-of-sight analysis between transmitters and receivers. As air movements are better understood with building obstructions known, the behaviours of pollution plumes can be studied in urban areas.

To generate a 3-D city model, accurate information about many features, such as DTMs, road networks, street blocks, buildings, trees and bridges, have to be acquired. Primary data to support feature extraction are aerial and satellite images and digital surface data collected by a variety of sensors. Airborne lidar can acquire digital surface data with high accuracy, fast turn-around time and reduced price. These characteristics have attracted much attention for various feature extraction tasks. While these tasks may be difficult or expensive to be completed when using traditional remote sensing data.

C.2.2 3-D Information Extraction

Lidar Expert can intelligently discriminate between terrain and non-terrain points, and have produced DTMs automatically using a hierarchical terrain recovery algorithm for typical areas with different terrain and coverage types. Some non-terrain points can be removed rapidly when the VSM is available, which is derived by thresholding the difference between first and last returns range data. Using intensity data can identify most pavements according to the reflectance properties and separability of various materials on ground. Thus, many hypothesized terrain points may be found easily.

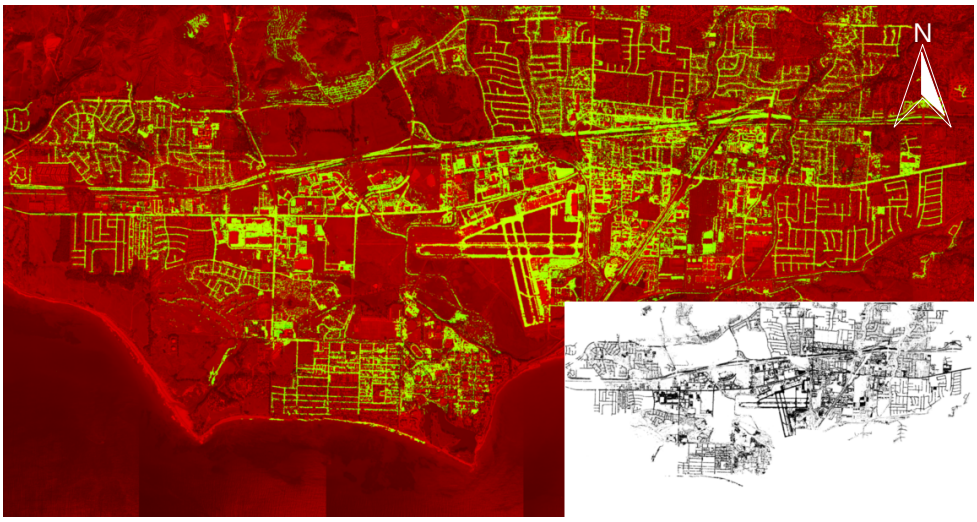


Figure C.1. Road ribbons overlaid on the DOQ (Santa Barbara airport)

Lidar Expert can reliably detect road ribbons and reduce the misclassification of the road class by integrating both the radiometry property of the intensity data and the height information of the DNM. The road ribbons detected for the Santa Barbara airport are

shown in Figure C.1, where the DOQ is placed on the red layer; the road ribbons are on the green layer; and at the right bottom part, it is a small scale picture of these road ribbons. In built-up areas where the road networks are arranged in grid format, a global grid constraint is used to formulate hypothesis for the presence of road primitives and to verify this by checking multiple criteria. The 3-D road segments and crossings hold the connectivity topology suitable for network analysis. The major attributes about each road segment including its width, length, and slope are also calculated.

Lidar Expert detects object footprints by segmenting the DNM and then refining them using morphological operations. The footprints of non-building objects (mainly trees) can be reliably deleted if the VSM is available. Building boundaries are distinguished from non-building ones by examining shape measures, and are modeled as rectangles, quadrangles, or regular polygons. Lidar Expert can produce regular shapes by enforcing the geometric regularity constraints such as orthogonality, parallelism and collinearity. Prismatic models are created for flat roof buildings by computing their average heights. Polyhedral models with vertical walls represent non-flat roof buildings. Many attributes, including orientation, height, width, length, and area, are calculated for each building.

Figure C.2. 3-D city model (Downtown Toronto)

C.2.3 Results

A case study is performed for downtown Toronto, which covers about 3.76 km^2 (see Section 4.2). Figure C.2 shows a screen shot of the 3-D city model visualized using ArcView GIS. The grid road model and prismatic building models are overlaid on the ortho-image draped on the lidar-derived DTM. The building shapes are classified into four types: rectangle, quadrangle, polygon and complex. The DTM is displayed at the transparency of 50%. The grid road model consists of 402 road segments and 302 road crossings. The area covered by the road class is 1.05 km^2, which is about 27.9% of the study area. The completeness and correctness of road segments are 84% and 91.6%, respectively. The area of all the building footprints amounts to 1.34 km^2, which is 35.6% of the study area. About 413 building models are created with the completeness and correctness around 92% and 97%, respectively.

C.2.4 Summary

The use of single- or multi-return range and intensity data enables the simultaneous usage of geometric and radiometric information for feature extraction. Lidar Expert is suitable for fast extraction of DTMs, roads and buildings.

C.3 PIPELINE SAFETY MANAGEMENT

C.3.1 Introduction

Demand for gas is being driven by growing worldwide demand for gas-fired electric power generation, and by growing industrial and commercial demands (NPMS, 2002). Natural gas distribution pipeline systems are being built, expanded, replaced, and planned worldwide (PLGI, 2002). Although pipelines are the best way to transport gas and energy resources, major excavation damage and natural disasters such as landslides and floods stress an emergency response organization’s abilities to plan and response. There are hundreds of natural gas and hazardous liquid pipeline accidents each year happened, resulting huge injuries, property loss and environmental damage (OPS, 2002).

As today's pipeline engineering and operations are more reliant on geospatial data for safety throughout a pipeline's life cycle of design, construction, operation, maintenance, emergency response, and post-disaster restoration, the rapid and cost-affordable acquisition of terrain data along the pipeline corridor becomes increasingly critical (NPMS, 2002; PLGI, 2002). Rapid and affordable acquisition of physical and ecological data along the pipeline corridor is critical for analyzing risks. Remote sensing technologies help identify risks to pipelines and provide cost-effective solutions.

Because of its narrow scanning swath in comparison to other sensors, airborne lidar is particularly suitable for rapid elevation data collection of pipeline routes in a cost-effective way compared with other elevation data collection systems (Hill et al., 2000; Optech, 2003), and should be considered for accurate pipeline mapping projects. The potential integration with other imaging technologies, such as optical, infrared, and SAR sensors, lidar data is expected to help pipeline risk managers solve safety problems and make decisions (Tao and Hu, 2002).

C.3.2 Pipeline Mapping and Risk Modeling

Pipeline safety problem starts before the actual construction begins and exists in the all life cycle of a pipeline. Since a pipeline has been built and placed into services, geologic hazards, corrosions and third-party damages all pose cumulative environmental and internal risks to its integrity (COB, 1999). In an effort to remove the guesswork from pipeline operations and reduce costs, new techniques have been employed to analyze pipeline risks and maintenance needs in a scientific fashion (NPMS, 2002; OPS, 2002).

Pipeline maps are produced to visualize the relationship among objects, and are initially generated to construct a pipeline (BTS, 2002). Gas and pipeline industry needs precise information when planning the most efficient and economical pipeline routes and quickly assessing the feasibility of construction in certain areas such as valleys between mountains. High-quality location maps are vital for safety reasons and in order to monitor any ground movement around the pipelines. During the construction of a pipeline, a route is established to go from the beginning of the pipeline project to the end of the pipeline

project. This procedure usually includes several steps including initial design, initial survey, pre-construction survey, construction survey, and as-built survey (BTS, 2002).

Lidar range data can be used to generate DTMs, produce tree canopy models if the data have multiple returns, and produce high quality orthoimages. Orthoimages can be used as maps to establish accurate geographic locations of pipelines. The aerial photographs can help to visualize accurate locations of pipelines with nearby environments, and may be used to extract breaklines after careful interpretation. The lidar intensity or thermal infrared images can be used to determine the 2-D location of the above or under ground pipelines, and are suitable for gas leakage detection according to the thermal anomalies. When multispectral images are available, different features may be distinguished by the supervised classification. Biomass prediction algorithms based on the NDVI and the SAVI also make use of these spectral bands. Moreover, improved biomass prediction is possible using the vegetation height data contained in multi-return lidar data.

However, lidar data may not replace photogrammetric mapping in some cases, for example, in the final design stage of the highway location and design process. Photogrammetric mapping is still required to produce highly accurate DTMs, as well as breaklines (Veneziano et al., 2002). So lidar can be used to collect large area or long corridors, providing designers with the terrain information enough to identify favourable alignments at planning stage. Once such alignments have been identified, detailed photogrammetric mapping can then be conducted for a lesser area or shorter corridor.

The penetration rate of laser beams mainly depends on the types of trees and season. Multiple returns allow the data to be analyzed and classified as vegetation, while the ground return allows DTMs of the bare ground to be generated and tree heights to be calculated (ALM, 2003; Optech, 2003). Lidar-derived DTMs can be used to calculate terrain parameters such as slope, aspect and curvature, and to produce DTM derivatives such as contours, profiles, watershed and drainage networks (Tao and Hu, 2002). The terrain parameters can help locate steep slopes, which could endanger the pipeline structure. The derivatives can identify segments having the highest risk factors.

To determine the high-risk segments of a pipeline from the viewpoint of natural disasters such as forest fire, some factors, such as sliding and fire risks, have to be considered. The sliding risk is subject to the slope. We can assume that a slope of 90^o is most dangerous, and a slope of zero has no danger. The fire risk is defined by the direction of prevailing winds, by the slopes that trend up from that direction, and by the vegetation biomass on those slopes. Slopes facing the prevailing wind and having high vegetation biomass are at high risk. Also, steeper slopes are at higher risk, as the flame front can travel quickly up the slope. So the pipeline risks are determined by slope, aspect, and biomass. Putting all these factors together, the risk rates can be estimated for each pipeline segments.

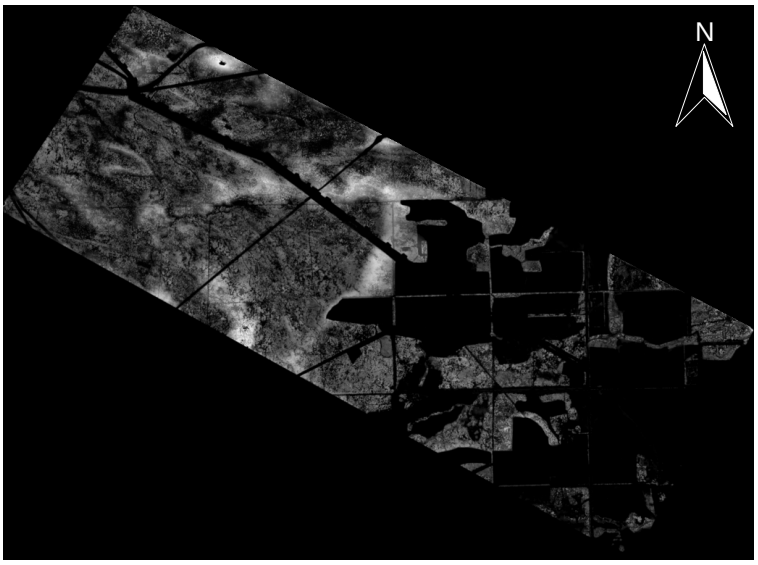


Figure C.3. The vegetation height model (Whitecourt)

Risk-rating maps can be created using a set of risk models by integrating the predicted risks due to fire, SCC, internal corrosion, surface erosion and terrain stability into one unified model (Tao and Hu, 2002). A risk-rating system along the pipeline would predicate potential trouble spots or estimate the potential risk posed by vegetation and features adjacent to the pipeline. This information could help maintenance supervisors better prioritize their work schedules to emphasize monitoring of the highest risk zones and assist risk management to assure safety in the pipeline's life cycle. For example, it can decide safer routes for placement of a new pipeline, and improve the fast response

ability in case of an accident (COB, 1999; NPMS, 2002).

C.3.3 A Risk Rating Example

The pipeline data was collected in August near Whitecourt, Alberta, using an ALTM 2050. The pipeline runs SE to NW through the centre of the ROI. The area contains of forested areas and a ridge crossing the northwest portion. The dataset consists of 20.9 million first and last returns lidar points covering a rectangular area of about 30 km^2. The study area has an elevation range from 733 m to 965 m. A patchwork of clear-cut was done to remove trees within about 42 meters along both sides of the pipeline's route. The range and intensity images converted from lidar data have a 1.2-m GSD resolution.

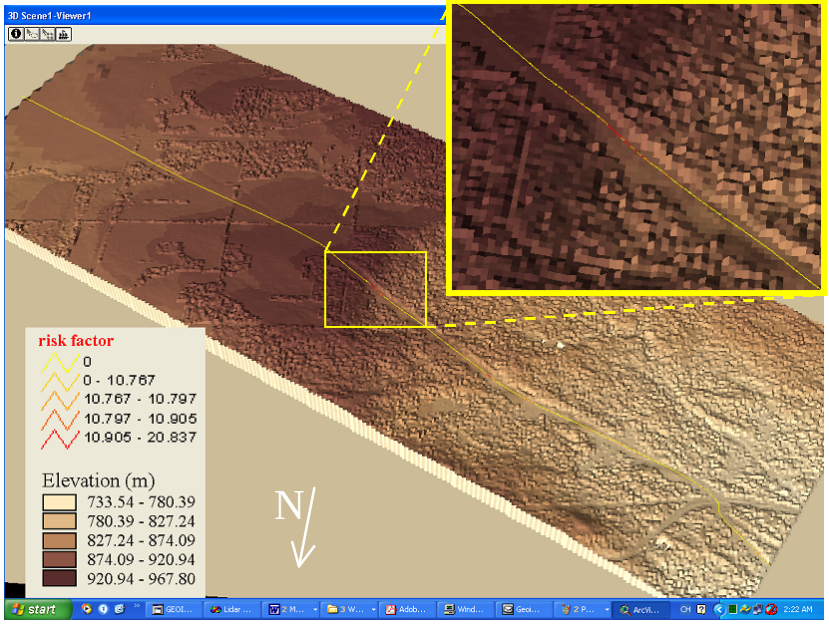


Figure C.4. Pipeline risk-rating map (Whitecourt)

The pipeline route is digitized on intensity image in 2-D according to the sketch map, and the vertex elevations are retrieved from the lidar-derived DTM using Lidar Expert. The terrain relief is small for most segments of the pipeline's route except the middle of the pipeline where the slope is about 17 degrees. The total risk index (RI) at a point of the pipeline is a linear combination of two risk factors, i.e.,

$$RI_{pipeline} = R_{slope} \cdot w_s + R_{biomass} \cdot w_b , \quad w.r.t. \quad w_s + w_b = 1 \qquad \text{(C.1)}$$

where R_{slope} is the slope risk; $R_{biomass}$ is the biomass risk; w_s (=0.6) and w_b (=0.4) are corresponding weights.

The slope risk is calculated along the pipeline route and is measured by the slope at that point. The vegetation heights are accumulated on the VHM along the perpendicular direction of the pipeline up to a depth of 100 m wide, and are then averaged to estimate the biomass risk. The VHM is a subtraction between the first-return lidar DSM and the derived DTM using the last-return lidar data as shown in Figure C.3, where a light gray stands for a larger height. Both risk factors are reclassified to a common scale, within the range of 0 to 25, giving higher scores to larger slopes and biomass. In Figure C.4, the 3-D pipeline is overlaid upon the lidar DSM, and is coloured by risk scores. Red segments are of high-risk, and may be damaged severely by a geologic hazard. The results show that lidar data contains plenty of information for estimating pipeline risks.

C.3.4 Summary

Safe pipeline transportation of energy resources is a major concern for the public. The virtual picture of a risk rating system to evaluate the risk levels of different segments of a pipeline route can be created by fusing information extracted from lidar data and other remote sensing data. This information can be used to improve the safe transmission of gas and liquid energy, and assist risk modeling to assure safety.

C.4 AIRFIELD SAFETY MANAGEMENT

C.4.1 Introduction

Today, there are more airplanes competing for airspace. To ensure the air safety, many factors have to be considered. The aeronautical community has recognized the acute need for accurate geospatial information in and around the airfield critical to flight safety, specifically for accurate runway positions, obstruction locations and heights, and topography around airfields. Knowing the location of an aircraft is only half of the solution. To bring an aircraft safely onto the runway with little else than satellite

navigation, the pilot will need very accurate and reliable geodetic coordinates for the landing runway. Currently, pilots are still expected to use conventional navigation aids or visual contact to direct the aircraft to the runway.

Airborne lidar is suitable for collecting accurate terrain data and providing feature information for airport safety management. In addition to identifying obstructions and designing approach procedures, pilots will be able to use the generated 3-D airfield models for flight training, pre-flight flythrough familiarizations, as well as increasing overall aircrew situational awareness relating to mission planning.

The airfield initiative document (AID) is a newly published specification for airfield obstruction identification (NIMA, 2001). It describes the use of 3-D OISs to survey glide slope obstructions, and puts new requirements for a safer flying environment. “The obstruction identification surface (OIS) consists of several surfaces with certain dimensions related to a specific runway approach, including primary surface (PS), approach surface (AS), primary/approach transitional surface (P/ATS), inner horizontal surface (IHA), conical surface (CS), outer horizontal surface (OHS), and conical/outer horizontal approach transition surface (C/OHATS)” (NIMA, 2001). When approaches share the exact surface, only one OIS is required. An obstruction is any object that penetrates an OIS, except where no obstruction penetrates the OIS; it shall be the highest object within the area. The obstructions are often extracted from photogrammetric and survey data. In addition, to avoid airport incursion, the surface of vehicular traverse ways (SVTW) is also needed (NIMA, 2001).

“The geospatial data required for obstruction identification around an airfield include the AEM, airfield features, and different combinations of the highest, the most penetrating, the highest approach and the highest non man-made obstructions/objects for analyzing each type of OIS surface” (NIMA, 2001). The AEM is in one arc second spacing each post having an absolute vertical accuracy of ±30.0 meters with respect to reflective surface. This vertical accuracy is required throughout the entire project area except for those posts that fall within the primary, primary approach, and primary/approach

transitional surfaces (NIMA, 2001). Airfield features, such as runway ends, must be surveyed to achieve high accuracy, for example, 0.3 m CE90 and 0.07 m LE90 for runway points. The required accuracy within OIS is lower. For example, within the PS, it is 6 m CE90 horizontally and 1 m LE90 vertically for the highest obstruction and the highest non man-made obstruction/object in each 912-m section of the primary area on each side of the runway. The required accuracies for other surfaces are normally lower.

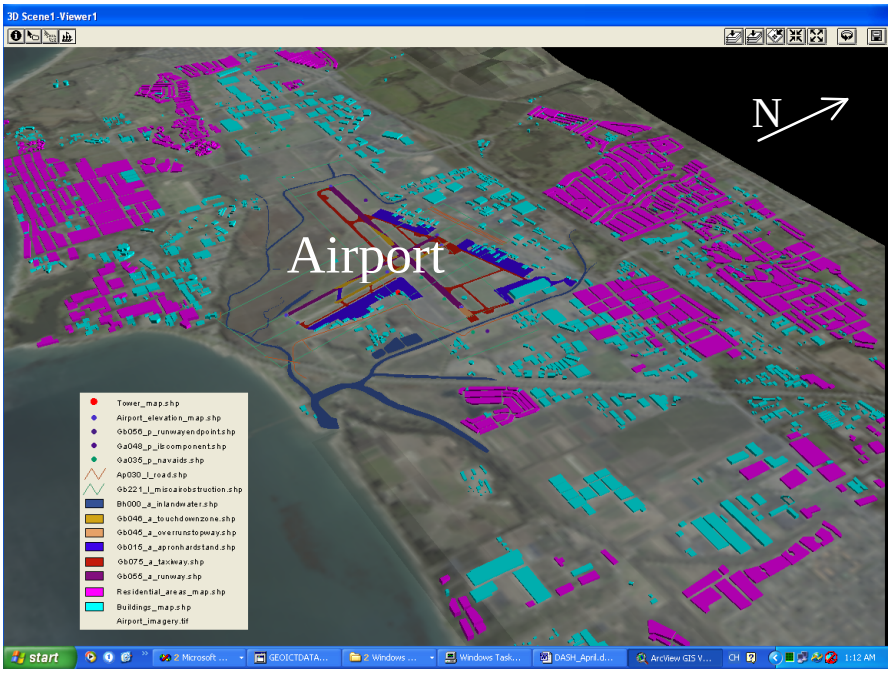


Figure C.5. 3-D airfield model (Santa Barbara airport)

C.4.2 Study Area

Santa Barbara airport has three runways (see Figure C.5). One is in east-west direction; another two are in the direction from southeast to northwest. At the north part, there are hills with varying heights. There are only a few large buildings around the airport, and most residential buildings are both smaller and lower than forested clusters. The airfield features are surveyed in field, including runway polygon and end points, taxiway, touchdown zone, overrun stop way, apron hardstand, buildings, roads, inland water areas, airfield elevation points, and many obstructions. The lidar dataset consists of 11.3 million first-return lidar points covering an area of 26 km^2. The terrain relief is from 0.31 m to 148.51 m. An aerial colour DOQ with a 1-m GSD covers the whole airfield.

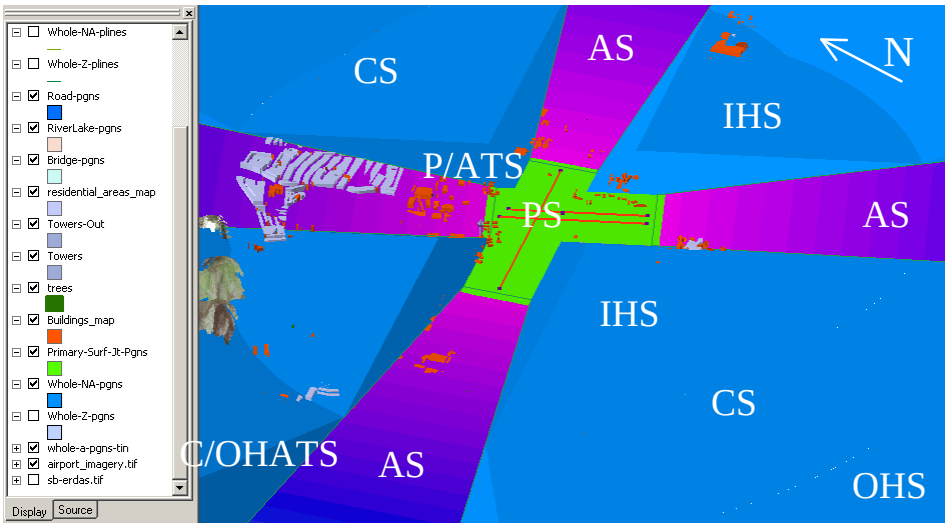


Figure C.6. Obstructions protruding OIS surfaces

C.4.3 Vertical Object Extraction

Lidar Expert derives a DTM with a 1.6-m GSD. Because last-return lidar data is not provided, it is impossible to reliably distinguish buildings from forested areas solely using shape measures. Some important objects, such as light poles, antennas, signs, small objects within airfield and large objects outside the HIS cannot be extracted using lidar data because of its limited resolution. So I manually digitize the 2-D outlines of 1,120 large buildings, 533 residential areas and forested areas around the airfield using the aerial orthophoto. A residential area is composed of multiple closely located houses with close roof heights. Lidar Expert then calculates the maximum object heights using object outlines, raw lidar points and the derived DTM. Some critical objects important for aviation protection, such as those sited on the runway and the control tower, are obtained in a same way. The 3-D airfield model is shown in Figure C.5, where 3-D features are extruded on the DOQ draped on the DTM. The measured features and extracted data include all the required data for the OIS analysis, and their accuracies are better than the accuracy required for obstruction identification.

C.4.4 Obstruction Identification and Risk-Rating

All the surfaces of two OISs are created following the AID as shown in Figure C.6. One runway has an OIS, and another two use a second OIS because their approaches share the

exact surface. All the digitized airfield objects that protrude the OISs are identified to support a safe flying environment (see Figure C.6). These identification results can help airport managers to check if their airfields meet the new safety requirements. To provide a clear view of the priorities of airfield obstructions to airport managers, the identified obstructions are classified into three risk categories by assessing risk index scores.

A RI is computed for each obstruction as a weighted sum of four risk factors, each corresponding to evidence upon which the risk evaluation is based. A larger RI score implies that the obstruction is more dangerous. The equation reads (Wang et al., 2003)

$$RI_{obstruction} = w_1 R_1 + w_2 R_2 + w_3 R_3 + w_4 R_4, \qquad w.r.t. \quad w_1+w_2+w_3+w_4=1 \qquad (C.2)$$

where R_i is the risk score of factor i ($1 \le i \le 4$); w_i is the weight of factor i. The higher the weight, the more influence a particular factor will have in the index model.

Table C.1. Risk levels of four risk factors

Risk levels	**Low**		**Median**	**High**	
	1	**2**	**3**	**4**	**5**
Distance (m)	< 2286	< 1000	< 600	< 300	< 100
Location	OHS, C/OHATS	CS	PS	IHS, P/ATS	AS
Type	Mountains	Residential houses	Objects within SVTW	Trees	Towers, buildings
Protrusion	N/A		OHS, C/OHATS	PS, CS	AS, IHS, P/ATS

Four risk factors are evaluated to take into account distance, location, type and protruding condition as described below:

- *Distance factor* R_1 is measured by the distance of an obstruction from the centreline of a runway. The risk levels are determined by the distance from the

HIS layer within 2,286 meters.

- *Location factor R_2* is measured by the position of an obstruction related to OIS. Different OIS layers are assigned with different risk levels.
- *Type factor R_3* is measured by the obstruction types, including buildings, trees, houses, mountains etc. For example, buildings are more dangerous than trees.
- *Protrusion factor R_4* is the measurement of the protruding condition. An obstruction protrudes any OIS has a risk level of no less than 3.

Table C.2. Pair-wise comparison of the relative importance of factors

Risk factors	**R_1**	**R_2**	**R_3**	**R_4**	**Row sum**	**Weight**
R_1	1	1/2	3	1/5	4.70	0.14
R_2	2	1	5	1/3	8.33	0.26
R_3	1/3	1/5	1	1/9	1.65	0.05
R_4	5	3	9	1	18.00	0.55
Sum					32.68	1.00

Each risk factor is assessed a score within the range of 1 to 5 as shown in Table C.1. The weights in Equation C.2 can be approximated by dividing the sum of values at that row by the total sum (i.e., the shaded cell in Table C.2). Saaty (1980) determined the weights using the analytic hierarchy process, which makes a series of pair-wise comparisons to determine the relative importance and ensures consistency between all the factors in a multi-criteria evaluation. In Table C.2, a pair-wise comparison matrix is constructed, where each factor is compared with the other factor, relative to its importance, on a scale from 1 to 9. The empirical values about the comparative importance between every two factors are shown in Table C.2. The weights are obtained by scaling the principal eigenvector of the matrix, that is, (0.13, 0.23, 0.05, 0.59). For example, the mountain summits sited at the northwest corner are about 1700 m from the HIS and protrude the C/OHATS (see Figure C.6). Their risk index is equal to 2.09 as calculated using Equation C.2 with weights substituted.

Saaty (1980) calculates a consistency ratio (CR) to check the probability that the ratings are randomly generated. The CR is defined by Equation C.3, where l_{max} is the principal eigenvalue of the matrix; *n* is the number of factors. A matrix with the CR value greater than 0.1 should be re-evaluated, and the process is repeated until the CR is less than this threshold. The CR is 0.0123 for the matrix in Table C.2.

$$CR = (l_{max} - n) / (n - 1) \qquad \text{(C.3)}$$

A risk level for each obstruction is assessed, and a part of the risk-rating map is shown in Figure C.7. The high-risk obstructions pose a severe threat to aircrafts and should be removed to conform to the AID. The median-risk obstructions may be kept, but should be inspected periodically.

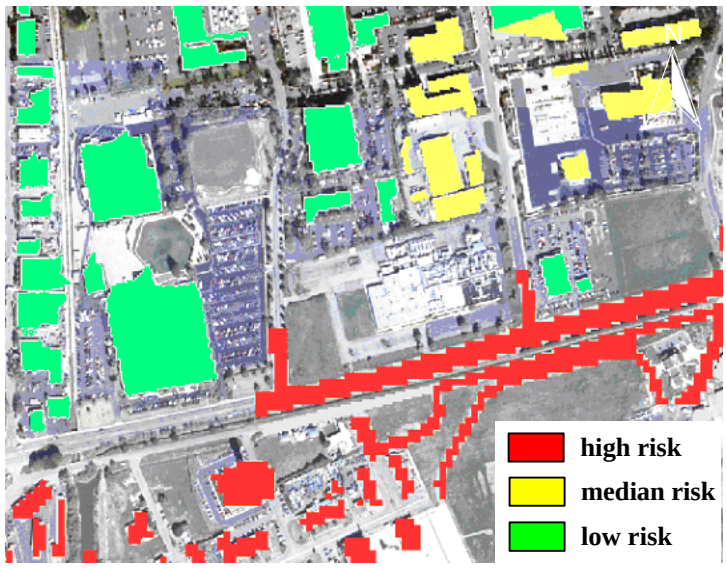


Figure C.7. Obstructions risk-rating map

C.4.5 Summary

This work makes a case study for obstruction identification following the new requirements described in the AID. By combining advanced lidar data processing techniques with traditional photogrammetric mapping services, new toolsets help airfield monitors solve problems and make decisions.