MODEL-BASED AUTOMATIC BUILDING EXTRACTION
FROM LIDAR AND AERIAL IMAGERY

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Suyoung Seo, B.S., M.S.

* * * * *

The Ohio State University

2003

Dissertation Committee:
Prof. Anton F. Schenk, Adviser
Prof. Alan Saalfeld
Dr. Beata Csatho

Approved by

Adviser
Graduate program in
Geodetic Science and
Surveying
© Copyright by
Suyoung Seo
2003
ABSTRACT

The automatic recognition and reconstruction of buildings from sensory input data is an important research topic with widespread applications in city modeling, urban planning, environmental studies, and telecommunication. This study presents integration methods to increase the level of automation in building recognition and reconstruction. Aerial imagery has been used as a major source in mapping fields and, in recent years, LIDAR data became popular as another type of mapping resource. Regarding their performances, aerial imagery has the ability to delineate object boundaries but omits much of these boundaries during feature extraction. LIDAR data provide direct information about heights of object surfaces but have limitations with respect to boundary localization. Efficient methods to generate building boundary hypotheses and localize object features are described. Such methods use complementary characteristics of two sensors. Graph data structures are used for interpreting surface discontinuities. Buildings are recognized by analyzing contour graphs and modeled with surface patches from LIDAR data. Building model hypotheses are generated as a combination of wing models and are verified by assessing the consistency between corresponding data sets. Experiments using aerial imagery and LIDAR data are presented. Three findings are noted: First, building boundaries are successfully recognized using the proposed contour analysis method. Second, the wing model and hypothesized contours increase the level of automation in building
hypothesis generation/verification. Third, the integration of aerial images and LI-
DAR data enhances the accuracy of reconstructed buildings in the horizontal and vertical directions.
Dedicated to my parents and my wife, Jae-Eun
ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my adviser, Prof. Anton F. Schenk, for teaching the theories and methods of Digital Photogrammetry and for making me aware of interesting research topics. Prof. Schenk’s instruction in Photogrammetry and Computer Vision has enlightened me and has led me to discover valuable ideas. I thank him for providing the great opportunity to work closely with him as a research associate during my years at The Ohio State University.

I would also like to thank Prof. Ayman F. Habib for his valuable lectures in Analytical Photogrammetry. His kind and enthusiastic teaching has brought me much joy.

Next, I wish to thank Dr. Beata Csatho, who extended my knowledge about remote sensing, in particular, the methodologies used to process hyperspectral images. I also appreciate her giving me a research position at the Byrd Polar Research Center when I needed additional financial support.

I thank Prof. Alan Saalfeld for his gentle and clear lectures in Map Projection and Spatial Data Structure. His teaching helped me to relax and enjoy the concepts he presented.
I acknowledge that I was fortunate to learn adjustment theory from Prof. Burkhard Schaffrin, who taught in elegant ways and also served as one of my general exam committee members.

I wish to thank Prof. Dean Merchant for giving a chance to work on a camera calibration project from USGS. I also wish to thank Dr. Grady Tuell, who provided a project from NOAA that enabled me to study at OSU as a full-time research associate.

My special gratitude goes to Dr. Dong-Cheon Lee, who greatly facilitated my coming to the USA as a research associate. I also thank Dr. Jay Hyoun Kwon, who energized my academic life; I enjoyed many talks and countless coffees with him. It would have been difficult to extend my study in the USA without the help of Dr. Dong-Cheon Lee and Dr. Jay Hyoun Kwon.

I would like to thank Prof. Manik Hwang, who loves students and encouraged me to extend my academic career. I wish to thank Prof. Yong-Il Kim, whose lectures in remote sensing I heard during my graduate school days in Korea. I wish to thank Prof. Byung-Guk Kim, who gave impressive adjustment lectures with gentleness. I wish to thank Dr. Dong-Hyun Kim, who guided a GPS study group and thus made the group aware of GPS technologies.

I would like to thank Jay J. Yoon, the president of Cadland Inc., and Joong-Hee Ryu, a representative director, both of whom gave me a valuable opportunity to experience GIS and remote sensing applications in the real world. I would like to express special thanks to Tae-Woo Jung, who was thoughtful and considerate toward me when I worked for the Cadland and Autodesk companies.

I would like to thank, Dr. Yuchan Yi, Keehwan Hong, Kee-Tae Kim, Sang-Ho Baek, Tae-Suk Bae, Shin-Chan Han, Chang-ki Hong, Gi-Cheol Ahn, In-Gyun Choi,
Jun-Pil Kim, Yushin Ahn, and Taehun Yoon who shared and enjoyed OSU campus life with me.

I wish to thank my colleagues in the Digital Photogrammetry Lab, Hsiang-Tseng Lin, Jack Cothren, Deeboon Methakullachat, Samuel Megenta, and Michel Morgan. I appreciate these individuals’ ready willingness to discuss ideas in the field.

From my deepest heart, I would like to express my gratitude to my parents for their endless love which manifests at any time for any reason. I am eager to thank my sister who takes care of my parents and my family. I would also like to thank my parents-in-law, who love our family with their whole hearts.

I thank my wife, Jae-Eun, and my daughter, Jihyun, who sacrificed their time and energy whenever I needed their help and who share the pleasures and pains of life with me.

Finally, my thanks go to God for staying with me all the time, wherever I am.
VITA

October 26, 1969 ......................... Born in Soon-Cheon, Korea

1988 ........................................ Dong-In High School, Pusan, Korea

1992 ........................................ B.Sc. Civil Engineering Urban Major,
Seoul National University, Korea

1994 ........................................ M.Sc. Civil Engineering Urban Major,
Seoul National University, Korea

Programmer, Cadland Inc., Korea

1997-1998 ......................... GIS/CAD Application Programmer,
Autodesk Inc., Korea

1999-present ......................... Graduate Research Associate, The
Ohio State University

PUBLICATIONS

Research Publications

Suyoung Seo, T. Schenk, “A study of integration methods of aerial imagery and
LIDAR data for high level of automation in 3D building reconstruction”. Multisensor,
Multisource Information Fusion: Architecture, Algorithms, and Applications VII, part

Suyoung Seo, “Data Fusion of Aerial Images and LIDAR Data for Automation of
Building Reconstruction”. 3rd International Workshop on Mapping Geo-Surficial

Suyoung Seo, B. Csatho, “Applications of Ice Penetrating Radar Data”. Technical
Report, Byrd Polar Research Center, The Ohio State University, 2002.

viii


**FIELDS OF STUDY**

Major Field: Geodetic Science and Surveying
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Abstract</th>
<th>ii</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>viii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xiii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xv</td>
</tr>
</tbody>
</table>

Chapters:

1. Introduction ........................................ 1
   1.1 General ......................................... 1
   1.2 Related disciplines .............................. 3
   1.3 Purpose of this study ............................ 4
   1.4 Outline of the dissertation ..................... 4

2. Background and Problems .............................. 7
   2.1 Introduction ..................................... 7
   2.2 Building recognition ............................ 10
   2.2.1 Single image driven recognition ............. 10
   2.2.2 Elevation driven recognition ................. 12
   2.2.3 Data fusion driven recognition ............... 14
   2.3 Building reconstruction ........................ 15
   2.3.1 Reconstruction from single images .......... 15
   2.3.2 Reconstruction from multiple images ......... 16
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2 Generation of contour graphs</td>
<td>60</td>
</tr>
<tr>
<td>5.4 Determination of building boundary contours</td>
<td>64</td>
</tr>
<tr>
<td>5.5 Experimental results of building recognition</td>
<td>69</td>
</tr>
<tr>
<td>6. Hypothesis Generation of Building Models</td>
<td>83</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>83</td>
</tr>
<tr>
<td>6.2 Segmentation of building roof surfaces</td>
<td>84</td>
</tr>
<tr>
<td>6.2.1 Generation of azimuth images</td>
<td>84</td>
</tr>
<tr>
<td>6.2.2 Generation of surface patches</td>
<td>87</td>
</tr>
<tr>
<td>6.2.3 Generation of surface patch adjacency graphs</td>
<td>89</td>
</tr>
<tr>
<td>6.3 Generation of wing models</td>
<td>90</td>
</tr>
<tr>
<td>6.3.1 Initiation of wing models</td>
<td>92</td>
</tr>
<tr>
<td>6.3.2 Determination of ending surface patches of wing models</td>
<td>94</td>
</tr>
<tr>
<td>6.4 Boundary generation of building models</td>
<td>97</td>
</tr>
<tr>
<td>6.5 Experimental results of building model generation</td>
<td>98</td>
</tr>
<tr>
<td>7. Hypothesis Verification of Building Models</td>
<td>108</td>
</tr>
<tr>
<td>7.1 Introduction</td>
<td>108</td>
</tr>
<tr>
<td>7.2 Verification with features from aerial images</td>
<td>109</td>
</tr>
<tr>
<td>7.3 Verification with features from LIDAR data</td>
<td>111</td>
</tr>
<tr>
<td>7.4 Experimental results of building model verification</td>
<td>112</td>
</tr>
<tr>
<td>8. Conclusions</td>
<td>126</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Approaches with respect to data and methods</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Terminology of elevation data</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Characteristics of sensors</td>
<td>27</td>
</tr>
<tr>
<td>3.2 Performance comparison between LIDAR and aerial images</td>
<td>27</td>
</tr>
<tr>
<td>4.1 Estimates of EOP parameters for left image</td>
<td>52</td>
</tr>
<tr>
<td>4.2 Estimates of EOP parameters for right image</td>
<td>52</td>
</tr>
<tr>
<td>4.3 Correlation of EOP parameters for left image</td>
<td>53</td>
</tr>
<tr>
<td>4.4 Correlation of EOP parameters for right image</td>
<td>53</td>
</tr>
<tr>
<td>5.1 Statistics of the data sets</td>
<td>69</td>
</tr>
<tr>
<td>5.2 Number of classified contours</td>
<td>74</td>
</tr>
<tr>
<td>6.1 Estimates of surface patch parameters</td>
<td>101</td>
</tr>
<tr>
<td>6.2 Estimates of the merged surface patch parameters</td>
<td>101</td>
</tr>
<tr>
<td>6.3 Anti-symmetry($\Lambda_k$) between the adjacent surface patches</td>
<td>104</td>
</tr>
<tr>
<td>6.4 Description of the generated wing models</td>
<td>104</td>
</tr>
<tr>
<td>6.5 Building corners generated by intersecting surface patches</td>
<td>106</td>
</tr>
</tbody>
</table>
7.1 Estimates of corners using features from aerial images . . . . . . . . . 121
7.2 Estimates of corners by integrating LIDAR data and aerial images . . 123
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Data acquisition and object reconstruction.</td>
<td>26</td>
</tr>
<tr>
<td>3.2</td>
<td>3D corner, wing and face models for feature grouping.</td>
<td>31</td>
</tr>
<tr>
<td>3.3</td>
<td>Framework of this study.</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>Relation between a 3D line and an arbitrary point in the line.</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>Implementation of the procedures for co-registration of aerial images.</td>
<td>46</td>
</tr>
<tr>
<td>4.3</td>
<td>Left and right aerial images for study area.</td>
<td>48</td>
</tr>
<tr>
<td>4.4</td>
<td>Distribution of control lines in perspective view.</td>
<td>50</td>
</tr>
<tr>
<td>5.1</td>
<td>Processing flow for building recognition using contour models.</td>
<td>56</td>
</tr>
<tr>
<td>5.2</td>
<td>Proper grid interval for feature extraction.</td>
<td>57</td>
</tr>
<tr>
<td>5.3</td>
<td>Height difference between target objects and background surfaces.</td>
<td>59</td>
</tr>
<tr>
<td>5.4</td>
<td>Closed((\hat{C})) and open((\tilde{C})) contours in a target area.</td>
<td>60</td>
</tr>
<tr>
<td>5.5</td>
<td>Ray crossing algorithm.</td>
<td>63</td>
</tr>
<tr>
<td>5.6</td>
<td>Contours and contour graph.</td>
<td>64</td>
</tr>
<tr>
<td>5.7</td>
<td>Comparison of boundary localization performance between filtering method and contour method.</td>
<td>65</td>
</tr>
<tr>
<td>5.8</td>
<td>Slope between contours.</td>
<td>66</td>
</tr>
</tbody>
</table>
5.9 Roof boundary contours ........................................ 68
5.10 Raw laser points in block L ..................................... 70
5.11 Raw laser points in block H ..................................... 70
5.12 Range images generated by TIN method and by nearest neighbor method. 72
5.13 Raw contours generated from the range image of block L .................. 75
5.14 Classified contours back projected into an aerial image of block L ...... 76
5.15 Hypothesized building contours in an exoskeleton image of block L .... 77
5.16 Classified contours for a building in an aerial image and in its exoskele-
ton image. ................................................................ 77
5.17 TIN model generated from raw laser points of block H .................... 78
5.18 Range image and contours of block H .................................. 79
5.19 Steep and non-steep contours of block H .................................... 80
5.20 Hypothesized building boundary contours. ................................... 81
5.21 Contours back projected in an exoskeleton image of block H ............. 82
6.1 Surface parameters and normal vector ..................................... 87
6.2 Surface patch adjacency graph (SPAG) ..................................... 90
6.3 Building modelling procedure using wings ................................... 91
6.4 Anti-symmetry of two surface patches ....................................... 92
6.5 Initiated wing models. .................................................. 94
6.6 Ending types of wing models ............................................. 94
6.7 Refined wing model after aggregating ending surfaces ..................... 95
CHAPTER 1

INTRODUCTION

1.1 General

In recent years, spatial information has become one of the most important information sources in human life. Because of rapid growing technologies in related fields, it is clearly expected that more information shall be associated with spatial information provided through systems such as the Geographic Information System (GIS).

Object reconstruction has been rigorously studied by many researchers in computer vision and photogrammetry. Because of the high capacity of computers, the information can be provided in more realistic ways for real world applications such as 3D city modeling, terrain analysis, tourism and other applications related to virtual reality. Hence, object reconstruction can be considered as one of the most needed technology to accomplish the virtual modeling paradigm for real world phenomena.

Advanced sensing technologies have been introduced to object recognition and reconstruction. Now, aerial photographs can be accurately processed in digital format. In recent years, the Ligh Detection And Ranging (LIDAR) scanning system became one of the promising data sources for reconstructing real world surfaces. Currently,
there are many other useful sensors for object recognition and reconstruction such as multispectral sensors, hyperspectral sensors and Synthetic Aperture Radar (SAR).

Because of the importance of buildings in human life and the complexity of building structures, building recognition and reconstruction have been studied rigorously. Many approaches have contributed fundamental concepts and pragmatic methods to the enhancement of automatic building recognition and reconstruction.

Despite extensive research in building recognition and reconstruction, however, the problem is still unsolved, at least as far as general and robust methods are concerned. One reason for this situation is that buildings have a wide variety of types and shapes depending on their purpose, culture, regions and countries.

The variety of building shapes and types sets a limit to the approach where a database with existing buildings is built which is then compared with building features extracted from sensors. A more promising approach seems to be in modeling buildings as polyhedral objects. Here, the reconstruction entails extracting building features, such as planar surface patches and edges, and then grouping them to higher ordered structures until enough evidence is accumulated to form building hypotheses.

In this study we describe a novel method of recognizing buildings and reconstructing them. Since both the recognition and the reconstruction, are ill-posed, we propose to use several sensors that yield complementary building information, and to further constrain the solution space by using appropriate domain knowledge.
1.2 Related disciplines

As state-of-art science and technology, computer vision and photogrammetry play key roles in object reconstruction. Shapiro and Stockman (2001) define the goal of computer vision as follows:

*The goal of computer vision is to make useful decisions about real physical objects and scenes based on sensed images.*

Photogrammetry, the major discipline in this study, is defined by the American Society of Photogrammetry and Remote Sensing (ASPRS) as follows:

*Photogrammetry is the art, science, and technology of obtaining reliable information about physical objects and the environment through the processes of recording, measuring, and interpreting photographic images and patterns of electromagnetic radiant energy and other phenomena.*

As the new generation of photogrammetry, digital photogrammetry is a comprehensive discipline that combines principles in computer vision and photogrammetry such as geometry, image processing, pattern recognition and object representation for object reconstruction and image understanding. Many important principles in digital photogrammetry are described in *Digital Photogrammetry* by Schenk (1999a).

This study exploits the advanced concepts in publications as follows:


- CVGIP: Image Understanding
1.3 Purpose of this study

The objective of the proposed study is to develop proper methods for building recognition and building reconstruction using aerial imagery and LIDAR data. In the study, new methods are proposed as follows:

- For building recognition, a method utilizing contours is suggested for detecting the initial boundary of building roofs. We establish contour graphs and analyze them in order to detect building roof contours as good approximations of building boundaries.

- For building reconstruction, we divide the reconstruction process into model generation and verification. In the model generation step, surface patches from LIDAR data are aggregated based on wing models as a new method of building modeling. In the verification step, hypothesized models are verified by integrating features from aerial images with the surface patches obtained from LIDAR data.

1.4 Outline of the dissertation

This dissertation is organized as follows.

Chapter 1 describes the general goal of and problems in building reconstruction and sets forth the purpose of this study.
Chapter 2 presents a review of the approaches used in building recognition and reconstruction. Because of the variety of approaches, they are reviewed in classified ways with respect to data types and methodologies. At the end of the chapter, the problems related to this research topic are stated and our approaches are proposed.

Chapter 3 presents the fundamental components considered for this study. The chapter describes the characteristics of aerial images and LIDAR data. Building models are described in terms of feature grouping and building representation. Methods of parameter estimation are described with respect to robustness and efficiency.

Chapter 4 describes a method used to align sensors to a common reference frame. Mathematical and stochastic models are described for the control line based orientation of aerial images. The estimates of the orientation parameters are presented along with accuracy information.

Chapter 5 presents a building recognition method based on contour analysis. The processes from contour generation to building boundary hypothesis generation are described in sequential order. Experimental results show the detected building boundaries from low and high density LIDAR data sets.

Chapter 6 describes a proposed method for building model generation. The chapter presents methods to estimate the plane parameters of 3D surface patches, group surface patches into wing models and combine wing models for a complete building hypothesis. At the end of the chapter, experimental results of building model generation are reported.

Chapter 7 presents a method to verify the hypothesized building model, which integrates features from aerial images and LIDAR data. The sensor models of aerial images and the correspondences between the generated model and the aerial image
features are expressed. Experimental results show a refined building model that integrates the building hypothesis and the extracted features.

Chapter 8 presents conclusions related to our proposed methods and suggests future works to extend the study.
CHAPTER 2

BACKGROUND AND PROBLEMS

2.1 Introduction

In recent years, there has been much research on building reconstruction in computer vision and photogrammetry. Due to the importance of buildings in many applications, numerous studies have investigated automating processes of building recognition and reconstruction. Their approaches and developments, however, are diverse. Because of the rapid improvement of computer performance, many processes in building reconstruction have been automated. Complex concepts and algorithms are introduced such as high level of data structures, fast processing of images and other fundamental components. State-of-art technology has improved processing performance and some processes are commercialized for automatic building recognition and reconstruction in robust and efficient ways.

In this chapter, we review the relevant research systematically in order to obtain a general overview of research trends and to identify the key achievements of the various approaches. The research can be divided into two broad topics - building recognition and reconstruction. To increase reconstruction performance, however, they can
Table 2.1: Approaches with respect to data and methods

<table>
<thead>
<tr>
<th>Factor</th>
<th>Building recognition</th>
<th>Building reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Optical images</td>
<td>Single image</td>
</tr>
<tr>
<td></td>
<td>Elevation data</td>
<td>Multiple images</td>
</tr>
<tr>
<td></td>
<td>Multispectral images</td>
<td>Elevation data</td>
</tr>
<tr>
<td></td>
<td>Hyperspectral images</td>
<td>Data fusion ...</td>
</tr>
<tr>
<td></td>
<td>Data fusion ...</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Region based recognition</td>
<td>2D line driven reconstruction</td>
</tr>
<tr>
<td></td>
<td>Edge based recognition</td>
<td>3D line driven reconstruction</td>
</tr>
<tr>
<td></td>
<td>Morphological operation ...</td>
<td>Surface patch driven reconstruction ...</td>
</tr>
</tbody>
</table>

be interlaced during production of buildings. Generally, the results of the building recognition process reduce the search space of features and the reconstruction process employs the features within the proximity of recognized regions for building model generation. However, the recognized buildings can be confirmed at relatively later stages by evaluating the consistency of the reconstructed buildings to the extracted features.

Research on building recognition and reconstruction, can be subdivided with respect to its data and methodologies. Table 2.1 presents the usual data types and methods for both tasks. Methodologies are greatly dependent on the data types they use because available features from the given data are keys to initiate and guide recognition and reconstruction in reliable ways. On the other hand, limitations in the data can be reduced by methods using domain knowledge such as epipolar constraints, possible building models and other useful knowledge about objects. We, therefore, employ methodology as the second criteria to review the papers.
Table 2.2: Terminology of elevation data

<table>
<thead>
<tr>
<th>Terms</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEM</td>
<td>Digital Elevation Model</td>
</tr>
<tr>
<td></td>
<td>a broad term including DTM and DSM</td>
</tr>
<tr>
<td>DTM</td>
<td>Digital Terrain Model</td>
</tr>
<tr>
<td></td>
<td>surface model consisting of only ground surfaces</td>
</tr>
<tr>
<td></td>
<td>not contains buildings, trees and other non-terrain surfaces</td>
</tr>
<tr>
<td>DSM</td>
<td>Digital Surface Model</td>
</tr>
<tr>
<td></td>
<td>physical surface in real world</td>
</tr>
<tr>
<td></td>
<td>includes buildings, trees and other objects</td>
</tr>
<tr>
<td>Normalized DSM</td>
<td>surface model generated by subtracting DTM from DSM</td>
</tr>
<tr>
<td></td>
<td>represents only non-terrain objects without terrain information</td>
</tr>
</tbody>
</table>

Regarding data types, aerial images and LIDAR data have been most often used because of their high accuracy and reliability. However, the characteristics and features of these data types are completely different. Optical features such as the radiometric boundaries in a scene are useful to hypothesize building outlines. On the other hand, elevation data are useful to detect surface discontinuities in the physical world. In addition, multispectral and hyperspectral images are widely used for automatic object recognition and for texture mapping of object surfaces.

The approaches are classified based on the actual data types rather than raw data types. For example, although some approaches use DSMs generated by matching multiple aerial images, we consider them to be elevation based approaches because the resulting spatial information provides height data rather than optical contrast. Table 2.2 presents the terminology associated with elevation data types using the concepts described in Baltsavias et al. (1995).
Now, we review approaches for building recognition and reconstruction with respect to the data sources and the general methodologies.

2.2 Building recognition

Many studies of building reconstruction have conducted building recognition at the initial stages. Approaches used in building recognition can be divided into three groups: optical feature driven approaches, elevation driven approaches and multisensor fusion methods.

2.2.1 Single image driven recognition

A single aerial image is the typical data type used in approaches based on optical features. Being based on radiometric contrast, most of these approaches extract primitive features such as edges and homogeneous regions. Relational graphs among the edges and regions are then established for high level processes.

Approaches based on single image features produce closed polygons to generate building hypotheses. In order to generate closed polygons, geometric and chromatic attributes of the primitive features are exploited together with their topological relations. As an important cue for recognition, buildings are mostly assumed to be simple flat types. Then, building outlines are generated by combining lines and regions. Because simple flat buildings are assumed to be recognized, buildings are reconstructed by extending recognized boundaries in vertical direction.

Many approaches employ line segments and their relationships in recognizing building regions (Lin and Nevatia, 1998; Collins et al., 1998; Kim and Nevatia, 1999;
Katartzis et al., 2001). In the approaches based on optical features, lines are aggregated together into closed polygons. Polygons are generated by combining collated line segments, which make parallelograms, and by considering the viewing geometry of camera models. To evaluate the likelihood of line combinations, Lin and Nevatia (1998) use certainty factors. They reduce the search space of the combinations by considering certainty factors rather than tracing feature relation graphs. Kim and Nevatia (1999) generate closed polygons based on neural networks or Bayesian networks. They use Bayesian networks designed to reflect evidence from natural dependencies. Katartzis et al. (2001) use a Markov Random Field (MRF) to find optimal relations between features. The relations are classified into supporting or competing relations between features based on the geometric and photometric attributes of the features.

For hypothesis verification, Lin and Nevatia (1998) exploit the shadows and walls neighboring the polygons. Collins et al. (1998) use other images to collect evidence. Through the geometry of camera models, the generated polygons are projected into other images and the features within the epipolar constraint zone are used for verification.

Brunn et al. (1996) use region features to hypothesize building regions by morphological operations. Levitt (1995) utilizes not only edges but also homogeneous regions to generate hypotheses of building regions. For verification and localization of buildings, Levitt (1995) performs a snake algorithm which takes the boundaries of homogeneous regions as initial outlines of buildings and integrates the geometry of edges and regions into the energy function of the snake model. However, some roof regions are not segmented well and are left undetected due to the low contrasts at
building outlines and the lack of evidence such as shadows and walls.

In summary, although the approaches based on single image features present rigorous perceptual organization, the building models are usually limited to simple shapes. Furthermore, some buildings do not seem to be detected and verified well because of the low geometric capacity of a single image.

2.2.2 Elevation driven recognition

Many studies have performed building recognition using elevation data because elevation features are more closely related to object parts than are optical features. In optical images, walls and shadows, as parts of buildings, are recognized through the camera model and through the mutual relations of features. On the other hand, in elevation data, they are detected more directly by surface segmentation. Until now, the majority of elevation data has been obtained from aerial images and LIDAR data. In aerial images, heights are computed by matching correspondent areas and features.

The methods for detecting high regions can be grouped as follows: 1. morphological operation followed by thresholding, 2. comparison of neighboring heights, 3. edge detection, 4. optimal configuration and 5. contour based method.

1. Morphological operators are used to detect high objects (Baltsavias et al., 1995; Eckstein and Munkelt, 1995; Haala and Hahn, 1995; Weidner and Förstner, 1995; Brunn and Weidner, 1998). The operator is set with a specific window size in order to suppress high objects in elevation data. To perform appropriately, the window size should be larger than the large buildings to be recognized. After
subtracting the filtered DSM or DTM from the original DSM, a normalized DSM is obtained and used to extract building regions by thresholding. The problem with this method, however, is the difficulty in selecting a proper window size.

2. Berthod et al. (1995) and Cord and Declercq (2001) extract regions where the height distribution is homogeneous and classify them into high and ground regions by comparing them with the neighboring regions.

3. Edges are used for building recognition in Baltsavias et al. (1995); Wang (1998). After extracting edges, they classify edges based on shapes such as moments. The main advantage of this method is that there is no need to know the size of structure elements as necessary in morphological operation. The limitation of this method, however, is that the building outlines may not be well detected due to missing edge pixels and small height differences between surfaces.

4. Brunn et al. (1998), Brunn and Weidner (1998) and Baillard et al. (1998) determine the optimal configuration of regions for building recognition. In this method, building regions are extracted by increasing the probabilities of predefined relations between regions. Brunn and Weidner (1998) classify the regions by Bayesian networks, where not only height data but also step and crease edges are exploited. Baillard et al. (1998) extract high blobs by grouping pixels based on the height condition and classify them into ground and non-ground segments by MRF optimization.

5. Baltsavias et al. (1995) make Multiple Height Bins (MHB) and utilize them from coarse to fine levels to detect buildings. Seo (2002) and Seo and Schenk (2003) use contour graphs to compute the slopes between contours. Contour
graphs are analyzed in order to extract building boundary contours. This method is described in detail in Chapter 5.

To summarize, the approaches based on elevation data present good approximations of building regions without generating many invalid hypotheses.

### 2.2.3 Data fusion driven recognition

In recent years, various types of data such as multispectral and hyperspectral images have been integrated with aerial images or LIDAR data for building recognition and reconstruction. Some of the data fusion approaches present rigorous classification of ground objects, in particular, by integrating DSM data and multispectral images.

Csathó et al. (1999) suggest a general strategy to recognize buildings using multisensor data sets that include aerial images, LIDAR data and multispectral images. McKeown et al. (1999) present two methods to detect building regions: the stereo based method and classification based method. The regions generated by classification of hyperspectral images or DSMs are used to prune the line segments which are not likely to be related to building boundaries. It is shown that both methods can significantly reduce the number of false positives. On the other hand, Tao and Yasuoka (2002) consider heuristic knowledge about the attributes of object surfaces with respect to sensors. Pixel values driven from color infra-red IKONOS images and DSM data are used together for this knowledge based classification. They compute the Normalized Difference Vegetation Index (NDVI) in order to exploit spectral knowledge of objects.

Haala and Brenner (1999) merge DSM and multispectral data at the pixel level. They apply ISODATA classification to the merged data. After classification, most
objects such as buildings, trees, and grass-covered areas are well recognized.

In summary, the results from fusion based recognition provide fruitful information about the object space, which is useful for object identification.

2.3 Building reconstruction

The two main tasks in automatic building reconstruction are to aggregate primitive features into higher levels of abstraction and to verify them in their topological and geometric aspects. The generated building models are verified and refined by exploiting the features related to the models. However, due to the incompleteness of the data features and the complexity of the building shapes, many approaches elaborate on the development of methods to generate building models with reliability and efficiency.

The approaches are reviewed with respect to data types. In addition, semi-automatic approaches are also described in order to understand how to integrate human knowledge and operation for efficient production of building models.

2.3.1 Reconstruction from single images

A single aerial image is typically associated with the approaches in this category. Because available features are limited to one image, these approaches rigorously exploit the object domain knowledge and organize features perceptually meaning structures (Lowe, 1987; Sarkar and Boyer, 1993).

Lin and Nevatia (1998) and McKeown et al. (1999) use walls and shadow casts in an image to predict the heights of buildings. They use the direction of illumination and camera geometry in order to compute building heights via geometric modelling.
For verification of the generated building models, edges from buildings and ground such as back vertical walls, edges on ground corners and shadows are regularly utilized.

Although these approaches have great merit in that three dimensional buildings can be automatically generated from a single two dimensional image, the building models are usually limited to rectangular flat buildings. In addition, the quality of walls and shadows in images seems to be sensitive to changeable factors such as camera geometry and illumination conditions.

### 2.3.2 Reconstruction from multiple images

Multiple aerial images have been one of the most popular data types for building reconstruction. Epipolar geometry inherent in overlapped images is substantially employed for grouping primitive features and for generation and verification of higher level abstractions. The productivity and the accuracy are usually dependent on the number of overlapped images.

Aerial images have proven to have a rich capacity to generate and verify complicated building hypotheses. Elaborate grouping mechanisms, however, are essential to retain robustness and flexibility during building model generation due to the incompleteness of features in the image domain and the complexity of buildings in the object space.

Roux and McKeown (1994) generate building models by evaluating the connectivity between corners and by combining line segments into 3D planes. They discard unreliable surfaces by evaluating radiometric consistency between primitive features. To reduce the combinatorial matching of corners, they exploit epipolar constraints, height ranges, corner types and corner directions. Collins et al. (1995) compute the
proper heights of buildings using a voting scheme, where height distribution of the
3D line segments is used to determine the optimal height. They represent building
shapes by parameters and estimate the parameters by introducing constraints such
as coplanarity and rectangularity.

Baillard et al. (1999) group 3D lines into 3D planes. 3D lines are refined after
matching by merging and growing. They use half planes which have rotation pa-
rameters about lines. The existence of the half planes is checked in multiple images.
The optimal angle of the plane is determined by checking the full range of angles
recursively. The half planes are grouped with other coplanar and collinear planes.

Willuhn and Ade (1996), Fischer et al. (1998) and Henricsson (1998) model
buildings in hierarchical ways to improve the performance of feature aggregation and
overcome the complexity of building shapes. Willuhn and Ade (1996) suggest a black-
board reasoning system, where buildings are modeled using knowledge about images,
features and house models. They divide reconstruction tasks into feature, structural
and conceptual levels. At the feature level, primitive features are extracted, their
relationships are established and 3D line segments are generated based on the feature
relationships. Next, at the structural level, 3D lines are grouped into surfaces based
on the rules of chromatic and photometric attributes. At the conceptual level, natural
characteristics of buildings such as roof ridge lines and symmetric relations are ap-
plied. Fischer et al. (1998) present well designed strategies and rigorous experiments
for automatic building reconstruction. They use corner models as anchors to aggre-
gate primitive features and generate higher levels of abstraction. Next, hypotheses
concerning building parts are generated based on the class of 3D corner models using
predefined relations among corner classes and building part models. The generated
building parts are assembled into complete building hypotheses by merging and gluing processes. Henricsson (1998) emphasizes that color attributes be one of the strong cues for grouping in addition to geometric attributes. Regions are grouped based on the similarity of color attributes and combines 3D line segments into 3D planes using geometric attributes. The generated 3D patches are finally intersected to delineate the boundaries of a building model and to establish the topological relations among the surface patches.

In summary, the approaches based on multiple images reconstruct relatively complex buildings in automatic ways. Nevertheless, the ambiguities associated with matching are not readily resolved and increase the number of the hypotheses to be generated and verified at each level of the reconstruction processes.

2.3.3 Reconstruction from elevation data

LIDAR data have rapidly become one of the main sources for building reconstruction in addition to DSMs from aerial images. Surfaces can be segmented using various methods such as clustering, edge detection, region growing and optimization. After finding surface patches, building outlines are delineated by the intersection of surface patches.

Brunn and Weidner (1998) detect the crease and jump type of edges and use the connected component labelling algorithm to group the homogeneous regions into planar surface patches. Based on the adjacent relations between the patches, they establish regularities such as parallel, symmetric and anti-symmetric among the surface patches. Maas and Vosselman (1999) presents two methods. One method is to compute building parameters using moments from point distribution and the other
method is to segment a point cloud into surface patches and intersect them for boundary delineation.

Weidner and Förstner (1995) exploit the boundaries from building recognition as initial data for building reconstruction. They convert the boundaries from images in vector format and simplify them by introducing the right angle constraint. The boundary generalization is optimized based on a Minimum Description Length (MDL) principle. This method is applied for building types such as parametric and prismatic models. Cord and Declercq (2001) use a Monte Carlo type simulation to group 3D points directly to surface patches. They set the number of patches and find the optimal configuration between points and planes by maximizing the specified probabilities. The roof types, however, are set to be rather simple shapes.

To summarize, it is shown that generic types of buildings can be modeled well by the intersection of surface patches. One critical problem, however, is that the building outlines are not determined precisely.

### 2.3.4 Reconstruction by data fusion

Various types of data such as aerial images, LIDAR data, ground plans, multispectral images and hyperspectral images are integrated together to increase the performance of automatic reconstruction and the accuracy of reconstructed buildings.

From topographic maps or GIS data, Brenner and Haala (1998), Haala and Brenner (1999) and Vosselman and Dijkman (2001) employ ground plans as the initial outlines of buildings. They generate potential surface patches from the ground plans and refine them by integrating LIDAR data. The ground plans are decomposed
into rectangular regions and points within the rectangles are clustered into several planar surface patches. They estimate plane parameters of patches using adjustment models such as Hough transform and least square methods. Generic roof structures are well reconstructed by splitting and merging surface patches and by combining the refined surfaces with the outlines from ground plans.

McKeown et al. (1999) exploit a class map from HYDICE images and to reduce time in evaluating the hypotheses generated from panchromatic imagery. The material types of the roof surfaces are identified by assigning textures from the class map.

Ameri (2000) uses DSM data and aerial images for automatic building reconstruction. Planar surface patches are extracted from DSM data and are intersected to generate B-rep building models. Then the building hypotheses are verified and refined using edge pixels in the aerial images. Sensor models of the aerial images and constraints among the building entities are exploited to estimate building boundaries.

In summary, approaches combining multisensor data, in particular, aerial images and DSM data, show the capacity to generate complex building models and to achieve good accuracies for the final products.

2.4 Semi-automatic reconstruction

Because of the complexity of building structures in the real world, it is useful to adopt human interpretation of building models. In automatic reconstruction, building recognition and modeling are the most crucial issues. Human operators, however, perceive and classify building shapes quickly and with high stability. Nevertheless,
some procedures such as collection of related features and computation of adjustment equations need to be performed by computers.

Gruen (1998) and Gruen and Wang (1998) present TOBAGO and CC modeler to generate complicated building models using the critical points in their systems. Critical points are measured by a human operator through visual inspection. The systems automatically analyze the distribution of critical points using predefined rules called parser and generate building models based on the result of the analysis.

Läbe and Gülch (1998) and Gülch et al. (1999) reconstruct buildings based on the input models and the points provided by a human operator. Within the proximity of the generated building models, linear features are collected and used to estimate the parameters of the primitive building models such as gables and hips. For robustness, a RANSAC type adjustment is applied during parameter estimation.

In Rottensteiner (2000), the system also requires a human operator to input the type of building models and corner points. Using the edge pixels close to the initialized building models, the building parameters are refined.

In summary, the semi-automatic approaches employ human interpretation in order to establish generic building structures and refine the geometry of the structures by using the features extracted from data.

2.5 Problem statement

As described in the previous sections, many useful methods have been developed for building recognition and reconstruction. However, efficiency and reliability of certain tasks can be improved by integrating multiple sensory data (Abidi and Gonzalez, 1992; Hall and Llinas, 2001).
To accomplish a high level of automation, we consider the following important factors.

For data sets

- Which data sets are appropriate to achieve accuracy and reliability in the tasks?
- If data sets have different reference systems, how should they be aligned?
- How do we integrate features from different data sets in efficient ways?

For building recognition

- If different types of the data sets are used, which data is more appropriate for building recognition?
- Which features are more appropriate to increase detection performance?
- Which properties of features are needed to recognize buildings?
- How can building outlines closely approximate physical building boundaries?

For building model generation

- Among the available features, which features are better for building model generation?
- When blunders are expected in the data sets, how do we estimate the parameters robustly?
- How should primitive features and intermediate structures be aggregated into models?
• How do we represent the generated building models?

**For building model verification**

• Which parameters of the generated models should be estimated?

• Which features from data sets are reliable for checking the topology and the geometry of the hypothesized models?

• How do we refine the generated building models?

• If different types of data sets are used, how can accuracy be improved by integrating features from different sensors?

**2.6 Proposed study**

Building recognition and reconstruction are rather complicated problems. Difficulties arise due to the complexity of buildings and missing features. To overcome these problems in the data domain, it is suggested that aerial images and LIDAR data be combined.

We suggest that inference channels be followed during recognition and reconstruction of buildings using a bottom-up or data-driven approach and that the reconstruction process be automated by adopting well-organized inference channels.

• We register the aerial images to the coordinate system of LIDAR data using control lines. In LIDAR data sets, control lines are obtained by intersecting two adjacent planar surfaces constituting crease edges that are likely to be observable in aerial images. Images are co-registered to LIDAR data with control lines by single photo resectioning.
• To hypothesize the occurrence of buildings, it is proposed that jump edges be detected by finding steep starting contours.

• Regarding automatic model generation, we use planar surface adjacency graphs which represent the adjacency among surfaces within boundary polygons. We suggest a new aggregation method to group the planar surfaces into a building model that is based on the wing model. Complicated roofs can be decomposed into several connected wing models.

• A wing model has two side planes which make the ridge line and two end planes which make the ending surfaces of wings whose type can be junction with other wings or hips or gables.

• The generated wing models are verified in aerial images by back projecting boundary lines. Using the height information from the boundary contour and the generated model, the back projected boundary can be approximated to the edges which exist in aerial images.

Geometry of the model is estimated during the hypothesis verification step. The topological relationships among features induced from the model play an important role in finding the optimal geometric locations of the model by removing blunders during the adjustment process. After model verification, the validated model has refined geometry whose coordinates are computed by considering LIDAR surfaces and aerial images simultaneously.
CHAPTER 3

BASICS OF THE PROPOSED STUDY

3.1 Data acquisition and building reconstruction

One of the ultimate goals in computer vision and photogrammetry is to reconstruct objects using sensed data. Figure 3.1 illustrates the relationship between data acquisition and object reconstruction. As can be seen, the goal of reconstruction is to describe objects at a certain level of abstraction. Hence, before we reconstruct the buildings, the proper abstraction level needs to be clarified. The abstraction can include various aspects of the buildings such as size, volume, shape, color and other properties.

In this study, the main goal is to describe building shapes. The shape of buildings can be represented using the geometric properties of entities and the relations of the entities. The entities of buildings can include boundary lines, surface patches and corners. However, those entities are complicated in the real world. For example, boundary lines are not exact straight lines, surface patches are not a perfect plane and corners are not a perfect intersection of lines and planes. Nevertheless, they can be simplified at certain levels of abstraction with reasonable tolerance. On the other
hand, the relations among the entities are crucial to achieve tight correspondences of the entities to the structures of the abstracted buildings in the real world.

### 3.2 Characteristics of sensors

Characteristics of sensors are important factors to decide the proper level of abstraction at which buildings need to be described. Table 3.1 summarizes the characteristics of aerial images and LIDAR data. From the characteristics, the data capacity and the potential error sources can be expected. Huisings and Pereira (1998) describe the systematic and random errors of LIDAR data with respect to the individual components of LIDAR system.

The characteristics of aerial images and LIDAR data are compared in Baltsavias (1999) and Schenk (1999b). Table 3.2 lists the performance of the two sensors in aspects of object recognition and boundary delineation. As can be seen, it is explicit that aerial images and LIDAR data have mutually complementary characteristics.
### Table 3.1: Characteristics of sensors

<table>
<thead>
<tr>
<th></th>
<th>Aerial images</th>
<th>LIDAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal</td>
<td>optic (visible)</td>
<td>range</td>
</tr>
<tr>
<td>Method</td>
<td>passive</td>
<td>active</td>
</tr>
<tr>
<td>Component</td>
<td>camera</td>
<td>GPS, INS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>laser scanning system</td>
</tr>
<tr>
<td>Ground coverage</td>
<td>continuous</td>
<td>not continuous</td>
</tr>
<tr>
<td>Dimension</td>
<td>2D</td>
<td>2.5D or 3D</td>
</tr>
<tr>
<td>Raw data</td>
<td>film, scanned images</td>
<td>list of points</td>
</tr>
<tr>
<td>Derivatives</td>
<td>ortho image, DSM</td>
<td>TIN, grid</td>
</tr>
</tbody>
</table>

### Table 3.2: Performance comparison between LIDAR and aerial images

<table>
<thead>
<tr>
<th></th>
<th>Aerial images</th>
<th>LIDAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advantages</td>
<td>Localize parts of object</td>
<td>Provide the existence of</td>
</tr>
<tr>
<td></td>
<td>boundaries</td>
<td>object surfaces at the</td>
</tr>
<tr>
<td></td>
<td>Relatively accurate</td>
<td>points after proper</td>
</tr>
<tr>
<td></td>
<td>in horizontal direction</td>
<td>boundary detection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>processing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Relatively accurate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>in vertical direction</td>
</tr>
<tr>
<td>Drawbacks</td>
<td>complicated to recognize</td>
<td>Have weak evidence to</td>
</tr>
<tr>
<td></td>
<td>objects</td>
<td>localize object boundaries</td>
</tr>
<tr>
<td></td>
<td>Relatively low vertical</td>
<td>Ambiguous delineation of</td>
</tr>
<tr>
<td></td>
<td>accuracy</td>
<td>object boundaries</td>
</tr>
</tbody>
</table>

In object recognition, LIDAR data provides relatively direct information about the existence of the objects in certain regions. On the contrary, regarding boundary delineation, in particular, for the jump edges, aerial images give more accurate features such as straight lines and corners at early stage of feature extraction.

As can be seen, the limitation of a data set from a single type sensor can be considered. In aerial images, there are occlusion areas where some features of buildings cannot be detected. Now, if we add other images from different views, occlusions
may be removed. But if the contrast at the ridge line is not good enough due to the high altitude of sun, those additional images may not provide enough information to extract sufficient features for reconstruction. Radiometric similarity at the edges is usually a big obstacle for optical feature based reconstruction in both interactive and automatic reconstruction. False features which do not come from the real object boundaries generate many invalid hypotheses. In LIDAR data, the localization of jump edges is problematic in man-made features. Because the scanning pattern is relatively irregular and not continuous, the detected edges are usually jagged.

Now, the fusion strategy is to exploit the advantages of each sensor while compensating for the disadvantages. The advantage of the aerial imagery is the ability to localize the object boundaries. The advantage of LIDAR data is the ability to distinguish objects by accurate height information. Height is useful characteristic for building recognition because building roofs usually have consistent heights at their eaves. Our contour analysis method for building recognition is based on this general fact. Surface patches from LIDAR data provide good clues to model building shapes because of their close correspondence to object surfaces in the real world. Hence, we exploit 3D surface patches for generating building model hypotheses.

### 3.3 Characteristics of buildings

Buildings can be decomposed into many individual components and properties (Fischer et al., 1999). Generally, buildings can be divided into two parts. One is the roof part and the other the wall part. Because the shape of walls usually conforms to that of roofs, most walls can be considered as the extensions of vertical planes that start
from roof eaves and end at ground surfaces. Hence, regarding building reconstruction, the roof part can be mainly considered. However, when the facade of buildings needs to be delineated, we may need to use other data sources such as terrestrial photogrammetric images and LIDAR data.

The building roof structures can be considered in two aspects, topology and geometry, as follows.

### 3.3.1 Topological aspect

In building reconstruction, topology is an important concept which ties the entities of buildings with specific relations. The relations include adjacency, connectivity, direction, inclusion and other relations. Potential relations between the entities are exploited to reduce the search space and to remove blunders at early stages.

Adjacency is frequently used for grouping features from data. Adjacency relations between features can be represented by adjacency graphs which link adjacent features using graph edges. The relations can be inferred from observations of the features. On the other hand, the relations can be predefined by building models. Surface patches in roof regions are adjacent to each other and those adjacencies can be recorded into adjacency graphs. Surface boundary lines can be generated from crease edges where two surface patches intersect. The lines can also be recorded with topological information such as their left and right polygons.

Data-driven processes use observed features to predict possible building models. This data-driven process increases the number of topological relations by adopting high level of relations inferred from the models. In contrast, a model-driven process
exploits the predefined model structures in order to find the corresponding features in the data.

### 3.3.2 Geometric aspect

Most man-made features have artificial characteristics like straight lines, symmetric planes, perpendicular or parallel line pairs and other artificial alignment of structures. As one of the man-made features, buildings have many artificial factors like parallelism, symmetry and orthogonality. The geometric relations can be tested among features ( Förstner et al., 2000; Heuel, 2001; Heuel and Förstner, 2001a) and exploited for generating constraints in building structures (Förstner, 1995). When buildings are described in generic models, the geometric aspects need not be considered. However, those geometric characteristics can be used to predict missing features during hypotheses generation and verification.

### 3.4 Models for feature grouping

Feature grouping is an essential process to generate building model hypotheses. During hypothesis generation, the features need to be well combined to represent the topology and the geometry of object structures in the real world. In addition, in order to enhance the performance of the hypotheses generation/verification processes, proper aggregation methods are crucial. Fischer et al. (1998) suggest aggregation models for grouping primitive features. Figure 3.2 shows the models in object space that can be used for grouping primitive features such as edges, corners and homogeneous regions. According to Fischer et al. (1998), a corner model is a vertex based aggregation and composed of a vertex, the lines and the regions adjacent to the vertex. Following their basic concepts, in our approach, wing models and face models
are defined as follows. A wing model is a two surface patch based aggregation and is composed of an intersection line of the surface patches, the lines, and the vertices adjacent to the surface patches. A face model is a surface patch based aggregation and is composed of the lines and the vertices adjacent to the surface patch. These aggregation models can be represented by graph data structures that describe adjacency relations between the aggregated features.

### 3.4.1 Corner model

A corner model can be initiated in various ways. From aerial images, a corner model can be generated by forward intersecting 2D corner points detected in image domain. In order to detect the corners, the features from corner operators or from intersection of straight lines are usually used. In LIDAR data, however, it is difficult to find corners reliably because edges in range images or in TIN models are mostly jagged.

Corner models are exploited to aggregate primitive features and to find remaining missing edges by narrowing the search space (Fischer et al., 1998; Lang, 2001). The corners are combined together when they are connectable based on the type of their
adjacent lines and regions. Using a group of the connected corner model, building parts are hypothesized by indexing plausible building part models among the pre-defined models. Although buildings are reconstructed well by introducing plausible geometric constraints inherent in building structures, the reconstruction process needs to check many invalid hypotheses caused by missing features, false matchings and the false indexing of building part models.

3.4.2 Wing model

Wing models can be initiated by grouping two anti-symmetric surface patches. The boundary lines of a wing model can be delineated by an intersection line of the side surface patches and the lines and the corners adjacent to the surface patches.

In wing model generation, the property of 3D surface patches and the relations among the surface patches are important to hypothesize building parts and to delineate the shape of the parts. The 3D surface patches can be generated by grouping 3D line segments from aerial images or by extracting homogeneous surface patches from LIDAR data. Here, when their point density is high enough, LIDAR data are more advantageous in extracting valid 3D surface patches.

After grouping 3D surface patches into wing models, wing models can be refined by adding other ending surface patches. This wing model has advantages because it can be used to generate building parts directly.

3.4.3 Face model

Face models can be initiated by a homogeneous surface patch or by grouping 3D line segments. The boundaries of a face model are delineated by aggregating its
adjacent lines and corners. When two face models are adjacent and anti-symmetric, they can be used to generate wing models.

The face model is useful to reconstruct buildings whose surfaces are not grouped into wing models. After generating building boundary hypotheses from LIDAR data, the boundary lines can be refined by integrating line segments extracted from aerial images.

3.5 Models for building representation

The shape of buildings can be represented in various ways. However, building models can be classified into three categories: polyhedral models, parametric models and CSG models as described in Braun et al. (1995) and Fischer et al. (1999). By adopting their concepts, the building models for representation can be described as follows:

3.5.1 B-rep

A Boundary Representation model (B-rep) is similar to polyhedral models, where the object surfaces are described by their boundary lines and surfaces. This model can represent generic types of buildings without geometric constraints among the entities. For grouping features, however, when there are some missing features in the data, the corresponding object structure may not be represented well.

3.5.2 Parametric model

A parametric model represents a building by a set of parameters such as length, width, height and other properties. This model can be used to extract buildings using semi-automatic methods, where it reduces the number of point measurement
times and also retains the geometric constraints in building structures. In automatic reconstruction, missing features can be predicted by some geometric constraints in the predefined model structures. However, the building shapes to be reconstructed are limited to certain types of predefined building models.

3.5.3 CSG model

Constructive Solid Geometry (CSG) represents a building as a combination of building parts. A complete building model is described by a tree structure, where the nodes represent the building parts and the edges the operations between building parts such as union, intersection and difference. The operations for CSG model generation is useful for grouping building parts into complete building models during building reconstruction.

3.6 Parameter estimation

3.6.1 Least square method

Since the least square method was introduced by Carl Friedrich Gauss and A.M. Legendre, the method has been used in a number of fields to estimate parameters. Mikhail (1976) and Koch (1999) describe rigorously how to apply the least square method to various situations.

The basic idea is to find a set of parameters which minimize the sum of the squares of the residuals, which can be written as:

$$\hat{\xi} = \arg \min_{\xi} \sum i r_i^2,$$

(3.1)

where $\xi$ is a parameter vector, $r_i$ is the $i$th residual and $\hat{\xi}$ is the estimate of $\xi$. The least square method fits the data properly to the mathematical model when
the observation data are randomly well distributed with respect to the model to be estimated. However, if data contains some significant blunders, it may not be detected properly and may produce wrong solutions as described in Fischler and Bolles (1981). Nevertheless, once blunders are removed by some robust estimation methods, the least square method is usually employed for their final estimation.

### 3.6.2 Trimmed mean

The trimmed mean method estimates parameters after discarding observations which produce relatively big errors or residuals. It sorts the residuals and discards the observations of the lowest and the highest parts of the residuals as described in Rice (1995). Then, the parameters are estimated using only the remaining observations. According to Rice (1995), the relation between the parameters and the residuals can be written as:

$$\hat{\xi} = \arg\min_{\xi} \sum r_{(i)}^2, \quad r_{(i)} \in \{ r_{\lceil n\alpha \rceil + 1}, \ldots, r_{n - \lfloor n\alpha \rfloor} \},$$

(3.2)

where $r_{(i)}$ denotes the $i$th residual after sorting, $\alpha$ the portion to be discarded and $\lfloor n\alpha \rfloor$ the floor of $n\alpha$. We exploit this method in order to estimate the plane parameters of local surface patches in Section 6.2.1.

### 3.6.3 Hough transform

Hough transform estimates parameters by using a voting scheme in parameter domain (Hough, 1959). Given observations, it computes all the possible parameter combinations for a set of observations and accumulates the occurrence of parameters in the discrete parameter space. Then, the solutions are determined by finding the maxima from the frequency distribution in the parameter space. Because of the
robustness against noise, it has been popularly used for estimating the parameters of features such as lines and planes in computer vision and image processing field. However, when the number of parameters increases, the size of parameter space multiplies combinatorially and the frequency distribution becomes sparse, which makes it difficult to detect the proper parameter sets. To cope with this problem, the parameters can be estimated in a sequential iterative way as described in Habib and Kelly (2001).

### 3.6.4 Random sampling

Fischler and Bolles (1981) and Rousseeuw (1984) presented random sampling methods which can discard blunders in robust and efficient ways. From the definition of Least Median of Squares (LMS) in Rousseeuw (1984), the optimal set of parameters are selected as:

$$
\hat{\xi} = \arg\min_{\xi} \text{median} r_i^2.
$$

(3.3)

This method selects randomly a set of observations and computes the corresponding parameters. Based on the model defined by the parameters, the residuals are computed for all the observations and their median is recorded as the representative residual of the parameter set. This procedure is performed repeatedly in proper number of times so that at least one parameter set should not be contaminated by blunders with a certain confidence probability. From the approach, RANdom SAmple Consensus (RANSAC) in Fischler and Bolles (1981), the number of iteration, $k$ need to be larger than the lower bound described as:

$$
k > \frac{\log(1 - G)}{\log(1 - (1 - \varepsilon)^u)},
$$

(3.4)
<table>
<thead>
<tr>
<th>Task</th>
<th>LIDAR data</th>
<th>Aerial images</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor co-registration</td>
<td>3D object lines</td>
<td>2D image points</td>
<td>EOPs of Aerial images</td>
</tr>
<tr>
<td>Building recognition</td>
<td>Contours and adjacency relations</td>
<td></td>
<td>Hypothesized building boundary contours</td>
</tr>
<tr>
<td>Building model generation</td>
<td>3D surface patches and adjacency relations</td>
<td></td>
<td>Hypothesized building models</td>
</tr>
<tr>
<td>Building model verification</td>
<td>2D image edge lines</td>
<td></td>
<td>Verified/refined buildings</td>
</tr>
</tbody>
</table>

Figure 3.3: Framework of this study

where $u$ denotes the number of parameters, $\varepsilon$ the contaminated portion among the observations and $G$ the probability of confidence that at least one of the solution is not contaminated. Finally, from Eq. 3.3, the optimal parameters are achieved by finding the parameter set among all the parameter sets which produces the least median.

We use this random sampling method to estimate the plane parameters of the segmented surface patches in roof regions in Section 6.2.2.
### 3.7 Framework of the proposed study

The general scheme of this study is described in Figure 3.3. As can be seen, the procedure is composed of four main tasks: co-registration, building recognition, building model generation and building model verification. The tasks can be organized in a proper order or fused together to obtain good products efficiently. The second column shows the major features which are exploited from LIDAR data and aerial images in order to perform the corresponding task and the third column the outputs after processing each task.

For sensor co-registration, 3D lines from LIDAR data and 2D points in aerial image are used as the corresponding features to align the data sets. Because in the given data set LIDAR data are referenced directly to object space and aerial images are not oriented yet, we exploit the 3D line features from LIDAR data set as the control features to estimate the Exterior Orientation Parameters (EOP) of aerial images. The oriented images have great capacity to describe object surfaces in terms of boundaries and textures. The 3D control lines are extracted by intersecting two planar surface patches. Because the 3D surface patches as the source of control features have influenced directly on the accuracies of subsequent tasks, the parameters of surface patches are robustly estimated by using a random sampling method described in Section 3.6.4.

For building recognition, contours are used as robust and efficient features to detect the occurrences of possible building regions. Building detection is performed based on contour graphs which represent the hierarchical adjacency relations between inner and outer contours. Building regions are hypothesized by introducing a domain
knowledge that building boundaries are steep in outward regions such as wall surfaces and relatively non-steep in inward regions such as roof surfaces.

For building model generation, wing models are exploited for grouping 3D surface patches. The grouping process is performed based on the topological and geometric relations among the surface patches. The grouping process is divided into two steps. First, wing models are initiated where the value of anti-symmetry between two adjacent surface patches is high. Second, the ending surfaces of the initiated wing models are determined based on the hierarchical graph which represents the adjacency relations among the initiated wings and the remaining surface patches. From the collected ending surfaces, the ending surface type of wing models are classified. A complete building model is generated by combining the wing models. The boundaries of wing models are delineated by 3D corner points where adjacent surface patches, wing models and the building contour detected in building recognition task intersect.

For building model verification, topology and geometry of the generated building models are verified and refined by integrating features from aerial images and LIDAR data. In order to combine the features, building models are back projected into aerial images through the sensor models determined in co-registration task. Edge lines in aerial images within proximity of the back projected building model are exploited to verify and refine the building models. Finally, the verified building models have good accuracy in both horizontal and vertical directions by integrating features from LIDAR data and aerial images.
CHAPTER 4

SENSOR CO-REGISTRATION

4.1 Introduction

A crucial aspect of integrating data from different sensors is to establish a common reference frame, a process known as sensor alignment or sensor registration (co-registration). Only when properly registered is it possible to compare features extracted from different sensors. We perform the registration with sensor invariant features (Schenk and Csathó, 2002).

For registration of data sets to a reference frame, it is necessary to employ a group of features which are commonly available over the different data sets. They can be obtained directly or indirectly from the data sets. However, it is important that the features from different data sets should represent common entities in object space.

In this study, we use a pair of aerial images which are not oriented and LIDAR data which are registered to object space. Hence, we use the features from LIDAR data as the control features to estimate orientation parameters of the aerial images. We use straight lines as control features which can be extracted from both data sets.
As a result, the aerial images are tightly aligned to the object space described by the LIDAR data.

The following is a description on how to use straight lines extracted from aerial images and LIDAR data sets to perform the orientation of aerial images with respect to the laser point cloud.

### 4.2 Models for co-registration

Co-registration is performed based on the mathematical model which describes the relations between the control lines in image domain and in object space. Figure 4.1 illustrates the geometric relations between the straight lines. Using the mathematical model, parameters are estimated by adjustment of the observations acquired from aerial images and from LIDAR data.

#### 4.2.1 Mathematical model for co-registration

As illustrated in Figure 4.1, the relationship between a point \( p(x, y) \) in image domain and the corresponding point \( P(X, Y, Z) \) in object space can be described by the collinearity condition as:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} =
\begin{bmatrix}
X_o \\
Y_o \\
Z_o
\end{bmatrix} + \lambda R_{\omega \phi \kappa}
\begin{bmatrix}
x - x_p \\
y - y_p \\
-f
\end{bmatrix},
\]  

(4.1)

where:

- \( f, x_p, y_p \): focal length and coordinates of principle points,
- \( \lambda \): scale factor of direction vector from camera center to object point,
- \( X, Y, Z \): coordinates of a point in object space,
- \( x, y \): coordinates of a point projected on camera coordinate system,
- \( X_o, Y_o, Z_o \): coordinates of camera exposure center,
- \( \omega, \phi, \kappa \): attitude angles of camera frame, and
Figure 4.1: Relation between a 3D line and an arbitrary point in the line
On the other hand, the condition that the extended point $P$ from images to object space lies on the 3D line $AB$ is described as:

$$
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
X_A \\
Y_A \\
Z_A
\end{bmatrix} + t \begin{bmatrix}
X_B - X_A \\
Y_B - Y_A \\
Z_B - Z_A
\end{bmatrix},
$$

(4.2)

where

$X, Y, Z$ coordinates of object points on the control line $AB$,

$t$ scale factor to extend the direction vector to arbitrary locations on the line,

$x, y$ coordinates of a point projected on camera coordinate system,

$X_A, Y_A, Z_A$ coordinates of one end point of a control line, and

$X_B, Y_B, Z_B$ coordinates of the other end point of a control line.

From Eq. 4.1 and Eq. 4.2, the relation between camera parameters and the coordinates of image and object points can be rearranged and we can estimate the EOP parameters from the observations. The equation is represented by related variables as:

$$
F = \left( \begin{bmatrix}
X_A - X_o \\
Y_A - Y_o \\
Z_A - Z_o
\end{bmatrix} \times \begin{bmatrix}
X_B - X_o \\
Y_B - Y_o \\
Z_B - Z_o
\end{bmatrix} \right) \cdot R_{\omega \phi \kappa} \begin{bmatrix}
x - x_p \\
y - y_p \\
-f
\end{bmatrix} \equiv 0.
$$

(4.3)

Eq. 4.3 can be rewritten in a simplified form as:

$$(V_1 \times V_2) \cdot V_3 \equiv 0,$$

(4.4)

where

$$
V_1 = \begin{bmatrix}
X_A - X_o \\
Y_A - Y_o \\
Z_A - Z_o
\end{bmatrix}, \quad V_2 = \begin{bmatrix}
X_B - X_o \\
Y_B - Y_o \\
Z_B - Z_o
\end{bmatrix} \quad \text{and} \quad V_3 = R \begin{bmatrix}
x - x_p \\
y - y_p \\
-f
\end{bmatrix}.
$$

Hence, Eq. 4.4 represents the coplanarity condition of the control lines in aerial images and LIDAR data.
4.2.2 Stochastic model for parameter estimation

The parameters to be estimated consist of \((X_o, y_o, Z_o, \omega, \phi, \kappa)\), the position and the attitude of the camera. Observations are the coordinates of the end points of the control lines, e.g. \(A(X_A, Y_A, Z_A)\) and \(B(X_B, Y_B, Z_B)\) and the image coordinates \(p(x, y)\) of points measured on the control lines in images. Because Eq. 4.3 is not linear with respect to the parameters, we linearize and estimate them iteratively.

After linearization, Eq. 4.3 can be rewritten using the Gauss-Helmert model (Koch, 1999) as:

\[
w = A\xi + Be, \quad e \sim (0, \sigma_o^2P^{-1}), \tag{4.5}\]

where

\[
w = F_o(X_o, Y_o, Z_o, \omega, \phi, \kappa, f, x_p, y_p, X_A, Y_A, Z_A, X_B, Y_B, Z_B, x, y),
\]

\[
A = \begin{bmatrix}
\frac{\partial F}{\partial X_o} & \frac{\partial F}{\partial Y_o} & \frac{\partial F}{\partial Z_o} & \frac{\partial F}{\partial \omega} & \frac{\partial F}{\partial \phi} & \frac{\partial F}{\partial \kappa}
\end{bmatrix},
\]

\[
\xi = \begin{bmatrix}
\Delta X_o & \Delta Y_o & \Delta Z_o & \Delta \omega & \Delta \phi & \Delta \kappa
\end{bmatrix}^T,
\]

\[
B = \begin{bmatrix}
\frac{\partial F}{\partial X_A} & \frac{\partial F}{\partial Y_A} & \frac{\partial F}{\partial Z_A} & \frac{\partial F}{\partial X_B} & \frac{\partial F}{\partial Y_B} & \frac{\partial F}{\partial Z_B} & \frac{\partial F}{\partial x_i} & \frac{\partial F}{\partial y_i}
\end{bmatrix},
\]

and

\[
e = \begin{bmatrix}
e_{X_A} & e_{Y_A} & e_{Z_A} & e_{X_B} & e_{Y_B} & e_{Z_B} & e_{x_i} & e_{y_i}
\end{bmatrix}^T.
\]

The improvement vector \(\xi\) is estimated as:

\[
\hat{\xi} = \left[ A^T(BP^{-1}B^T)^{-1} \right]^{-1} A^T(BP^{-1}B^T)^{-1}w \tag{4.6}
\]

and the dispersion of the estimated parameters is computed by:

\[
D\{\hat{\xi}\} = \sigma_o^2[ A^T(BP^{-1}B^T)^{-1} A ]^{-1}. \tag{4.7}
\]

The error vector is predicted as:

\[
\hat{e} = P^{-1}B^T(BP^{-1}B^T)^{-1}(w - A\hat{\xi}). \tag{4.8}
\]
Using the predicted error vector in Eq. 4.9, the a posteriori variance component is estimated as:

$$\hat{\sigma}^2 = \frac{\tilde{e}^T P \tilde{e}}{n - m},$$

where $n$ is the number of observation equations and $m$ is the number of parameters.

In order to terminate the iteration (Mikhail, 1976), convergence is tested by comparing the magnitude of the variance components in Eq. 4.9 as:

$$\left| \frac{\hat{\sigma}_{oi+1} - \hat{\sigma}_{oi}}{\hat{\sigma}_{oi}} \right| < \epsilon,$$

where $\hat{\sigma}_{oi}$ and $\hat{\sigma}_{oi+1}$ are the $i$ th and the $i + 1$ th estimated variance component, respectively and $\epsilon$ is a convergence threshold.
4.3 Implementation of co-registration

We implemented co-registration using data sets of aerial images and LIDAR data. Figure 4.2 shows the flow of implementation procedures. As illustrated in the figure, the correspondent lines are collected from both data sets. Then, the features are used as observations in the adjustment model described in Section 4.2.2. The data set covers urban areas in Ocean City in Maryland. Detailed information about the data set is described in Csathó et al. (1998). A pair of black and white aerial images were obtained by scanning aerial photographs in 24 µm. Figure 4.3 shows the left and the right aerial images used for co-registration. LIDAR data are composed of point
clouds acquired by about ten overlapped missions. The point density of LIDAR data is about 1.2 points/m².

In the given data sets, roof ridge lines are found to be the appropriate features as control features.

From the LIDAR data, 3D lines are extracted by intersecting planar surface patches. Surface parameters are estimated in a robust way by using a RANSAC based adjustment. For determination of 3D control lines, two adjacent surface patches are intersected to generate two end points of the control lines. The 3D control lines can be well described using two points, where the points have accuracy information from error propagation of the surface parameters.
Figure 4.3: Left and right aerial images for study area
From aerial images, the correspondent lines are collected by measuring points on the ridge lines of roofs. As described in the mathematical model, the 2D points have to be on the control lines but do not have to directly correspond to the end points of the control lines.

Figure 4.4 shows nine 3D control lines collected by intersecting two adjacent surface patches in LIDAR data. In aerial images, two end points are measured on the image line which corresponds to a 3D control line. The accuracy of measured points is set to be one pixel, 24 \( \mu \text{m} \) in adjustment process.

On the other hand, EOP parameters are also estimated by control points which are measured based on visual inspection of the laser point distribution. We collect points which seem to be close to the corners of foreground objects. To increase the visual contrast, points are color coded according to their heights. Considering the density of LIDAR data and the height distribution of points within proximity, the measurement accuracy is expected to be 1.0 meter in the horizontal and 0.5 meter in the vertical. The corresponding points are also measured in aerial images. The measurement accuracy in aerial images is expected to be 24 \( \mu \text{m} \).

Table 4.1 and Table 4.2 show the estimated EOP parameters using the control point method (set A) and the control line method (set B). As can be seen, there are noticeable differences in EOP estimates between two implementation results. As an extreme case, for example, in Table 4.1, \( X_o \) in set A is about 6m different from the estimate \( X_o \) in set B. Hence, it confirms the importance of control lines to register aerial images to LIDAR data. In order to verify EOP estimates obtained from the control
Figure 4.4: Distribution of control lines in perspective view
line approach, the control lines are back projected in aerial images. As a result, back projected control lines in aerial images match correctly to the corresponding image features.

However, some eccentric results can be seen in the standard deviation. As listed in Table 4.1 and Table 4.2, the standard deviations of EOP parameters are relatively large compared to usual cases. In addition, estimates of the parameters are highly correlated as shown in Table 4.3 and in Table 4.4.

The differences from the usual accuracy are due to limited control features. For accurate estimation of orientation parameters, it is necessary that the control features are distributed well over the image. However, as shown in Figure 4.3, the available features are limited to land area. Furthermore, the control lines are limited further to the reliable building ridge lines.

Hence, when the special limitations of the data sets are considered, the results seem to be reasonable. In addition, although the accuracy of EOP estimates have relatively high standard deviations and correlations, the parameters produce accurate estimation of object features using the corresponding features as shown in Table 7.1. Thus, we can conclude that the aerial images are appropriately registered to the object space represented by LIDAR surfaces.
Table 4.1: Estimates of EOP parameters for left image

<table>
<thead>
<tr>
<th>EOP parameter</th>
<th>Estimated value - set A</th>
<th>Estimated value - set B</th>
<th>Standard deviation - set B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_o$ (m)</td>
<td>492376.359</td>
<td>492370.783</td>
<td>1.193</td>
</tr>
<tr>
<td>$Y_o$ (m)</td>
<td>4242024.696</td>
<td>4242021.407</td>
<td>0.990</td>
</tr>
<tr>
<td>$Z_o$ (m)</td>
<td>542.499</td>
<td>536.852</td>
<td>0.488</td>
</tr>
<tr>
<td>$\omega$ (deg)</td>
<td>-0.7169</td>
<td>-0.0855</td>
<td>0.0976</td>
</tr>
<tr>
<td>$\phi$ (deg)</td>
<td>-0.0729</td>
<td>-0.2648</td>
<td>0.1024</td>
</tr>
<tr>
<td>$\kappa$ (deg)</td>
<td>73.9337</td>
<td>74.2329</td>
<td>0.0428</td>
</tr>
</tbody>
</table>

Table 4.2: Estimates of EOP parameters for right image

<table>
<thead>
<tr>
<th>EOP parameter</th>
<th>Estimated value - set A</th>
<th>Estimated value - set B</th>
<th>Standard deviation - set B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_o$ (m)</td>
<td>492464.976</td>
<td>492449.705</td>
<td>0.948</td>
</tr>
<tr>
<td>$Y_o$ (m)</td>
<td>4242326.686</td>
<td>4242323.403</td>
<td>0.553</td>
</tr>
<tr>
<td>$Z_o$ (m)</td>
<td>540.713</td>
<td>538.845</td>
<td>0.544</td>
</tr>
<tr>
<td>$\omega$ (deg)</td>
<td>-0.7169</td>
<td>-0.4795</td>
<td>0.0568</td>
</tr>
<tr>
<td>$\phi$ (deg)</td>
<td>-0.0729</td>
<td>-0.4564</td>
<td>0.0999</td>
</tr>
<tr>
<td>$\kappa$ (deg)</td>
<td>73.9337</td>
<td>74.5412</td>
<td>0.0179</td>
</tr>
<tr>
<td></td>
<td>$X_o$</td>
<td>$Y_o$</td>
<td>$Z_o$</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>$X_o$</td>
<td>1.0000</td>
<td>0.5292</td>
<td>0.2491</td>
</tr>
<tr>
<td>$Y_o$</td>
<td>0.5292</td>
<td>1.0000</td>
<td>0.9384</td>
</tr>
<tr>
<td>$Z_o$</td>
<td>0.2491</td>
<td>0.9384</td>
<td>1.0000</td>
</tr>
<tr>
<td>$\omega$</td>
<td>-0.4707</td>
<td>-0.9972</td>
<td>-0.9616</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0.9993</td>
<td>0.4978</td>
<td>0.2136</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>0.9837</td>
<td>0.6713</td>
<td>0.4193</td>
</tr>
</tbody>
</table>

Table 4.3: Correlation of EOP parameters for left image

<table>
<thead>
<tr>
<th></th>
<th>$X_o$</th>
<th>$Y_o$</th>
<th>$Z_o$</th>
<th>$\omega$</th>
<th>$\phi$</th>
<th>$\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_o$</td>
<td>1.0000</td>
<td>0.5936</td>
<td>-0.9761</td>
<td>-0.6044</td>
<td>1.0000</td>
<td>0.9529</td>
</tr>
<tr>
<td>$Y_o$</td>
<td>0.5936</td>
<td>1.0000</td>
<td>-0.7517</td>
<td>-0.9999</td>
<td>0.6006</td>
<td>0.8075</td>
</tr>
<tr>
<td>$Z_o$</td>
<td>-0.9761</td>
<td>-0.7517</td>
<td>1.0000</td>
<td>0.7606</td>
<td>-0.9780</td>
<td>-0.9930</td>
</tr>
<tr>
<td>$\omega$</td>
<td>-0.6044</td>
<td>-0.9999</td>
<td>0.7606</td>
<td>1.0000</td>
<td>-0.6113</td>
<td>-0.8153</td>
</tr>
<tr>
<td>$\phi$</td>
<td>1.0000</td>
<td>0.6006</td>
<td>-0.9780</td>
<td>-0.6113</td>
<td>1.0000</td>
<td>0.9554</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>0.9529</td>
<td>0.8075</td>
<td>-0.9930</td>
<td>-0.8153</td>
<td>0.9554</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Table 4.4: Correlation of EOP parameters for right image
5.1 Introduction

Several approaches for building recognition use height information that is obtained by matching aerial images or by interpolating LIDAR points into a regular grid (Balt-savias et al., 1995; Berthod et al., 1995; Eckstein and Munkelt, 1995; Haala and Hahn, 1995; Baillard et al., 1998; Brunn and Weidner, 1998; Wang, 1998). In these studies, a crucial issue is to localize the optimal boundaries of potential buildings as the initial approximations of building roofs.

The generation of DTMs from aerial images is problematic in homogeneous regions. The preferred approach with LIDAR data is to interpolate a regular grid. This may result in significant variations of the accuracy in local regions, particularly in situations where the point density is coarser than the grid posts.

An alternative approach is to use contours for estimating initial building boundaries. Here, the critical issue is to determine boundaries from contours as a good approximation of jump edges between buildings and non-building surfaces.

We use a graph data structure of contours to aid the automatic recognition process. Graphical representation of contour model is useful to analyze topography (Freeman

54

In this paper, we use a contour model approach as a kernel resource for recognizing buildings in the object space (Seo, 2002). Figure 5.1 show the scheme of the proposed approach.

5.2 Generation of range image from LIDAR data

As a preliminary step, grids need to be generated by interpolating irregularly distributed point data at regularly spaced posts. There are two main factors that influence the outputs: grid interpolation method and interval.

First, regarding the interpolation method, grids can be generated based on TIN models or on nearest neighbor points. Because TIN based methods exploit local triangles generated from triangulation, the accuracy of the heights within the local triangles is directly impacted by structures of the triangles. In general, the quality of the triangle structure can be assessed by evaluating their correspondences to the real surfaces. Hence, denser the point distribution is, more likely the local triangles represent well the surfaces in real world. On the contrary, when the density is very low, most of the triangles cover bigger areas than they can represent. Thus, we employ weight factors based on the distances when using low density points.

Second, range images are generated by interpolating raw LIDAR data at regularly spaced grid posts. Because LIDAR points are irregularly distributed and the point
Figure 5.1: Processing flow for building recognition using contour models
density is variant over the regions, it is important to determine a proper interval between grid posts. Figure 5.2 illustrates the relations among LIDAR points and the objects to be recognized. As can be seen, objects should be recognized with their approximate boundaries using proper methods. In order to detect the buildings without loss of information, it is important to maintain the contents in the original data during interpolation.

To achieve this condition, we apply a sampling theorem which defines the upper bound of grid interval as:

\[ D_s < \frac{D_d}{2} \]  \hspace{1cm} (5.1)

where \( D_s \) is the sampling distance and \( D_d \) is the average distance between raw data points.

Average distance is computed using the area of the region and the number of points in raw data. However, specific regions such as rivers which do not reflect laser
signals need to be subtracted from the entire region to compute the actually scanned region. On the other hand, we also need to consider the efficiency of the data set by reducing the resulting image size while maintaining the features in raw data. After all, considering the sampling theorem and the practical aspects, we find the proper average distance between grid posts as:

\[ D_s < \frac{D_d}{2} = \frac{1}{2} \left( \frac{A_{\text{entire}} - \sum A_{\text{invalid}}}{N} \right)^{\frac{1}{2}} \]  

(5.2)

where \( A_{\text{entire}} \) is the area for the entire region and \( A_{\text{invalid}} \) is the area for null data regions.

### 5.3 Surface representation with contour graphs

#### 5.3.1 Extraction of contours

Contours delineate object surfaces with identical height lines. Hence, contours can be considered as a type of the extracted features from surface data. Contours can be generated from TIN or grid data which represent the object surfaces. The shape of contours can be divided into two types depending on their closedness. One is non-closing contours which are open contours within a study area. The other is closing contours which connect itself. If target area is large enough, most of the high objects can be extracted as closed contours. From the general characteristics of urban objects, we can assume that the contours are not overlapped.

Some factors need to be considered in the extraction of contours. Extraction of contours can be considered as a process generating discrete surfaces at a regular interval in the vertical direction. As the distance between grid post, the vertical interval need to be properly chosen to optimize the processing speed and the performance to determine contours close to the real building boundaries. Figure 5.3 shows the
relations between object heights. As can be seen, the top of a building is usually higher than its background objects like roads and parking lots.

From the sampling theorem, the maximum vertical interval is computed by dividing the minimum difference of heights to be detected by two. Based on the domain knowledge about buildings, the minimum height difference is also considered to reduce the number of contours to be generated.

Thus, we can make a rule how to determine the proper vertical interval as:

$$d_v < \frac{\min (h_o - h_b)}{2}$$

(5.3)

where $h_o$ is the height of target objects and $h_b$ is the height of background objects.
Figure 5.4: Closed(\(\mathcal{C}\)) and open(\(\overline{\mathcal{C}}\)) contours in a target area. Closed contours and open contours are denoted by solid and dotted lines, respectively.

### 5.3.2 Generation of contour graphs

As shown in Figure 5.1, the generation of contours from range images is followed by arranging them into a graphical representation, called a contour graph. Figure 5.4 illustrates the contour shapes after contour extraction process. Based on their closedness, the contours are classified into two groups as follows:

\[
\mathcal{C} = \{C_i| i = 1, 2, \ldots, N\} = \mathcal{C} \cup \overline{\mathcal{C}},
\]

where \(\mathcal{C}\) is the set of all contours, \(\mathcal{C}\) the set of closed contours, and \(\overline{\mathcal{C}}\) the set of open contours. As usual, \(\cup\) denotes the exclusive union of two sets.

In our approach, from Eq. 5.4, closed contours are exploited to detect boundaries of the buildings. The closed contour set is represented as:

\[
\hat{\mathcal{C}} = \{C_i| IsClosed(C_i) = True, i = 1, 2, \ldots, N\}
\]

where \(IsClosed\) is an unary predicate stating that a contour is closed.
We define the contour graph, $G_{\text{contour}}$ as a set of nodes, $G_{\text{node}}$, containing closed contours and a set of edges, $G_{\text{edge}}$ that represent the adjacency between pairs of contours by:

$$G_{\text{contour}} = \text{Graph}(G_{\text{node}}, G_{\text{edge}}).$$  \hspace{1cm} (5.6)

For detecting closed boundaries, only closed contours are used.

The directional adjacency between closed contours is used for the necessary relationship between contours in order to establish the hierarchical representation of contours. The direction is determined by checking whether a contour encloses another one or is enclosed by another contour. Thus, we can represent nodes and edges of a contour graph as:

\[
\begin{align*}
(G_{\text{node}}) &= \hat{C} \\
(G_{\text{edge}}) &= \{(\hat{C}_i, \hat{C}_j)|\text{EnclosedBy}(\hat{C}_i, \hat{C}_j) = \text{True} \text{ and } \text{Adjacent}(\hat{C}_i, \hat{C}_j) = \text{True}\},
\end{align*}
\hspace{1cm} (5.7)
\]

where $\text{EnclosedBy}(\hat{C}_i, \hat{C}_j)$ is a directional predicate stating that $\hat{C}_i$ is enclosed by $\hat{C}_j$. Similarly, $\text{Adjacent}(\hat{C}_i, \hat{C}_j)$ is a mutual predicate expressing that two contours are directly adjacent to each other.

The inclusion relationship between two contours can be reduced into the inclusion test of point-in-polygon because if a point $P_i$ as a vertex of contour $\hat{C}_i$ is within the other polygon defined by points in $\hat{C}_j$, $\hat{C}_i$ is always enclosed by $\hat{C}_j$. This, of course, under the assumption that the surfaces from range images are not overlapping each other. The following statement expresses the reasoning mechanism:

\[
\text{EnclosedBy} \left(\hat{C}_i, \hat{C}_j\right) \leftrightarrow \text{EnclosedBy} \left(\exists P_i, P_i \in \hat{C}_i, \hat{C}_j\right). \hspace{1cm} (5.8)
\]

This inclusion test is only performed if the area of contour $\hat{C}_i$ is less than the area bounded by $\hat{C}_j$, as it is impossible for a smaller contour to enclose a bigger one.
Furthermore, we can skip the inclusion test if the height of $C_i$ is considerably larger than that of $C_j$, indicating objects with a significant vertical dimension. Only cases are tested for inclusion if the following condition holds:

$$\left( \text{Area}(\hat{C}_i) < \text{Area}(\hat{C}_j) \right) \land \left( \text{Height}(\hat{C}_i) > \text{Height}(\hat{C}_j) \right),$$

where $\text{Area}$ and $\text{Height}$ are functions to obtain the area and height of the argument contour.

Point-in-polygon relations can be tested by winding number or ray crossing algorithms (O’Rourke, 1998, p. 239-245). We prefer ray crossing for efficiency reasons.

Figure 5.5 illustrates the ray crossing method to test if contours enclose other contours. The algorithm counts the number of points where a half of the infinite straight line starting from a point of a potential inner contour crosses a potential outer contour. The relations are determined based on the number of intersection points. In Figure 5.5, the number of intersection points on a contour $C_A$ crossed by a half line starting from a point $P_B$ on another contour $C_B$ is one, that is, odd. In case of the contour $C_C$, the number of intersection is four, that is, even. The rule can be written in a functional equation as:

$$\text{EnclosedBy}(\hat{C}_i, \hat{C}_j) = \begin{cases} 
\text{True} & \text{if } \text{NumberOfIntersections} \left( \exists P_i, P_i \in \hat{C}_i, \hat{C}_j \right) = \text{odd} \\
\text{False} & \text{otherwise.}
\end{cases}$$

where the function $\text{NumberOfIntersections}(P_i, C_j)$ counts the number of intersection points on a contour $C_j$ crossed by a half line starting from a point $P_i$. From Eq. 5.6 and Eq. 5.10, if $\text{NumberOfIntersections}(P_i, C_j)$ is odd, a directional edge starting from the node $\hat{C}_i$ to the node $\hat{C}_j$ is recorded into a contour graph.
Figure 5.5: Ray crossing algorithm

Figure 5.6 depicts eleven closed contours that are stored hierarchically in the contour graph. This is accomplished by performing the test EnclosedBy for all reasonable pairs satisfying the conditions defined by Eq. 5.9. In the figure, contour 1 and 4 are the outer most contours that enclose 2 and 3 contours 2 and 3. Contour 11 is highest and is enclosed by contour 10.

For generation of contour graphs, the procedure is summarized as following steps.

1. Check closedness of the contours and select only closed contours.

2. Compute the area of all closed contours.

3. Select two contours for an inclusion test.

4. Check the area and height conditions between two contours as described in Eq. 5.9.
5. If the conditions above are valid, then implement ray crossing to find the intersection points.

6. If the number is odd, then store the inclusion relation in contour graphs.

7. Repeat steps 3–6 for a new pair of contours.

5.4 Determination of building boundary contours

Building boundaries are characterized by significant height changes. From the established contour graphs, the height change can be determined robustly by evaluating the slopes between adjacent contours. The contour model method can use circumjacent areas of object boundary to be extracted. This is in contrast to local filtering methods that use only pixels within predefined window. Thus, boundary localization is superior to local filtering methods as it is virtually impossible to determine an optimal filter size due to varying object boundary shapes. Moreover, gaps in edges
Figure 5.7: Comparison of boundary localization performance between filtering method(left) and contour method(right). An object to be detected is illustrated by a dotted line. In left figure, an orange rectangle denotes a 5 by 5 filter and red dots show edge pixels that may be detected by the local filter. In right figure, orange lines represent the contour lines around the object and the red line is the expected line to be determined by contour analysis.

...may exist if the response of the edge operator is weak. The proposed contour method avoids these disadvantages.

Figure 5.7 illustrates two methods - filtering and contour analysis. As can be seen, for the filtering method, any predefined filter size may not be optimal to all the objects to be detected because of the variation in object sizes and shapes. On the contrary, in contour method, the performance is not dependent on the filter size and the real building boundary are closely localized together with height information.

From the previous section, after establishing contour graphs, the contours have a set of the properties as:

\[ \hat{C}_i = (\{P\}, \text{height, area, perimeter, } \{\text{InnerContours}\}, \{\text{OuterContour}\}). \quad (5.11) \]
Perimeters and areas of contours are stored as the essential properties of contours for subsequent processing. Now, the extracted contours are classified first into steep and non-steep contours. Here, the steepness of the contours are determined by using the areas and the perimeters of the contours themselves and their inner contours. Because our purpose for contour processing is to detect building outline contours, we discard contours whose areas are smaller than minimum area criteria for building contours.

In the contour model, the degree of height changes are evaluated by average slopes between adjacent contours as illustrated in Figure 5.8. The average slope of contour $C_i$ toward higher direction is computed as:

$$AveSlope(C_i) = \tan \bar{\alpha} = \frac{\Delta h}{W},$$  \hspace{1cm} (5.12)
where $\bar{\alpha}$ is the average angle, $\Delta h$ is the height difference, and $\bar{W}$ is the average width between a contour $C_i$ and its inner contours, respectively. In Eq. 5.12, average width can be computed using the areas and perimeters of the contours as:

$$\bar{W} = \frac{\Delta A}{L}, \tag{5.13}$$

where the area between contours, $\Delta A$ is

$$\Delta A = \text{Area}(C_i) - \sum \text{Area}(\text{InnerContours}(C_i))$$

and the average perimeter, $L$ is approximated as:

$$L \simeq \frac{1}{2} \left( \text{Perimeter}(C_i) + \sum \text{Perimeter}(\text{InnerContours}(C_i)) \right).$$

Hence, the average width in Eq. 5.12 is computed using the areas and perimeters of the contours as:

$$\bar{W} \simeq \frac{2[\text{Area}(C_i) - \sum \text{Area}(\text{InnerContours}(C_i))] \text{Perimeter}(C_i) + \sum \text{Perimeter}(\text{InnerContours}(C_i))}{\text{Perimeter}(C_i) + \sum \text{Perimeter}(\text{InnerContours}(C_i))}. \tag{5.14}$$

Finally, building boundary contours are extracted as vertical discontinuities of surfaces depending on whether a contour is steep or not. The contours are classified by introducing a slope threshold value. Because the slopes in the building roofs are usually not as steep as the walls, we can choose a proper threshold value of the slope which can distinguish roof contours from wall contours. Thus, contours can be classified into two groups - steep and non-steep contours.

As illustrated in Figure 5.9, boundaries of high objects can be properly approximated by the contours of which inner slopes are low and outer slopes high. Using
this approach, it is also possible to extract multiple boundaries in vertical direction as can be seen in Figure 5.9.
<table>
<thead>
<tr>
<th>Data set</th>
<th>No. points</th>
<th>Distance $dX$ (m)</th>
<th>Distance $dY$ (m)</th>
<th>Area $(m^2)$</th>
<th>Density (points$/$$m^2$)</th>
<th>Height $Z_{min}$ (m)</th>
<th>Height $Z_{max}$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>859</td>
<td>118</td>
<td>118</td>
<td>13987</td>
<td>0.0747</td>
<td>-38.36</td>
<td>-25.45</td>
</tr>
<tr>
<td>H</td>
<td>11358</td>
<td>138</td>
<td>128</td>
<td>17788</td>
<td>1.0653</td>
<td>-36.47</td>
<td>-19.54</td>
</tr>
</tbody>
</table>

Table 5.1: Statistics of the data sets

5.5 Experimental results of building recognition

Because point density of the LIDAR data is one of the most crucial factors influencing the performance of object recognition, we select two types of data sets whose densities are fairly different each other. Hence, the experiment will show if our approach is applicable to the general cases.

Figure 5.10 and Figure 5.11 show the views of two data sets whose point densities are fairly different each other. Table 5.1 provides general characteristics of the data sets with respect to the dimensions of the blocks in horizontal and vertical directions. As can seen, the areas of two sets are similar but the number of points are remarkably different. The point density of block L is more than ten times lower than the density of block H. In fact, the points of block L are acquired from one single laser scanning mission but the points of block H are a set of points from about ten overlapped missions. As shown in Table 5.1, due to the very low density of point distribution and the relatively small building heights in block L, it may be difficult to detect buildings using usual filtering and region growing methods due to the low point density and the small height differences. Region growing methods may have problems to obtain the proper building boundaries. Hence, building recognition using this low density data can be considered as an extreme challenge to any approaches.
Figure 5.10: Raw laser points in block L

Figure 5.11: Raw laser points in block H
Figure 5.12(a) shows a range image generated by using regular TIN method which interpolate linearly the heights using three corner points in every triangle. As can be seen, because of some improperly triangulated structures, in particular, along the boundaries of buildings and trees, artificial effects can be explicitly noticed.

On the other hand, a nearest neighbor method produced an appropriate result. Figure 5.12(b) shows the range image generated by interpolating four nearest points at each grid post. To produce this range image, heights are weighted in inverse proportion to square of the distances in order to reduce the interpolation errors at jump edges. As can be seen, the interpolation method is important to increase the performance of building recognition.

Although the interpolation method based on nearest neighbor points shows a better result than the TIN based method, this method causes artificial undulations of the surfaces, in particular, in planar surfaces. Hence, in high point density data, linear inverse weighting based on TIN model will be better to generate range images.

Figure 5.17 demonstrates that a TIN model with a dense point distribution can represent well the surfaces in real world. As can be seen, many triangles are constructed well along the boundaries of buildings and trees. Hence, in this case, TIN based interpolation is more reasonable. As expected, the range image shown in Figure 5.18(a) describes appropriately the real surfaces.

The generated range images are used to extract contours in a regular vertical interval. Because buildings are usually at least two or three meters higher than the neighboring ground surfaces, we set the height interval to 0.5 meter. The interval can
Figure 5.12: Range images generated by TIN method and by nearest neighbor method. (a) TIN method. (b) Nearest neighbor method with inverse distance square weighting.
be set smaller but it increases the number of contours to be extracted and so makes the subsequent processes delayed.

Figure 5.13 and Figure 5.18(b) show the resulting contours with 0.5 meter height interval. As can be seen, most of the high objects such as buildings and trees are well enclosed by the contours. In addition, it can be noticed that buildings are surrounded by many steep contours which are mutually close in the horizontal direction.

The criteria for steepness is set to 1.0 by introducing domain knowledge about buildings. Because most buildings have roofs whose slopes are less than the criteria, this value seems to be suitable to differentiate building roof contours from other contours such as wall contours.

Figure 5.14 presents the classified contours back projected in an aerial image of block L. Figure 5.19 also shows the classified contours with respect to their slopes for block H. As can be seen, contours within roof regions can be well differentiated from the other contours based on the boundary contours.

Figure 5.20(a) and Figure 5.20(b) show the hierarchical relations between the detected boundary contours. Using this relations, the detected regions can be refined further. For example, in Figure 5.20(a), contour 1 can be refined by subtracting points in contour 11.

The hypothesized building contours can be verified using other features in LIDAR data or in aerial images such as textures. Figure 5.15 and Figure 5.21 present the hypothesized building boundaries in exoskeleton images, where the exoskeleton images show gray scale image features such as corners, homogeneous regions, straight lines and non-straight edge regions (Fuchs and Förstner, 1995). For example, Figure 5.16
shows a valid building hypothesis. As can be seen, the hypothesis can be verified the correspondent features in aerial images. On the other hand, false hypotheses such as contour 1, 6 and 14 in Figure 5.15 and contour 6 in Figure 5.21 can be discarded because their textures in the aerial images are not homogenous. Furthermore, changes in the real world can be detected. The contours 2 and 16 in Figure 5.15 and contour 1 in Figure 5.21 show discrepancies from the features in aerial images. Hence, they can be classified into new objects such as trees.

Table 5.2 summaries the number of contours which are generated and classified for building detection. As can be seen, although the number of all the contours is fairly different in the two data sets, the number of steep and non-steep contours is small in both data sets. Hence after classifying contours, the subsequent process can be performed in fast ways. The extracted boundary contours are used as one of the important surface patches to generate building models.
Figure 5.13: Raw contours generated from the range image of block L.
Figure 5.14: Classified contours back projected into an aerial image of block L.
Figure 5.15: Hypothesized building contours in an exoskeleton image of block L.

Figure 5.16: Classified contours for a building in an aerial image and in its exoskeleton image.
Figure 5.17: TIN model generated from raw laser points of block H.
Figure 5.18: Range image and contours of block H. (a) Range image. (b) Contours.
Figure 5.19: Steep and non-steep contours of block H.
Figure 5.20: Hypothesized building boundary contours. (a) 2D view. (b) 3D view.
Figure 5.21: Contours back projected in an exoskeleton image of block H.
CHAPTER 6

HYPOTHESIS GENERATION OF BUILDING MODELS

6.1 Introduction

Hypothesis generation of building models is one of the most important tasks in automatic building reconstruction because of the complexity of buildings and the incompleteness of features. During hypothesis generation, buildings are modeled in the aspects of geometry and topology. The key factors to generate proper hypotheses are topological consistency of the structure of objects and geometric closeness to the entities of objects. Hence, robust feature extraction and well designed grouping mechanisms are crucial to achieve those key factors.

In photogrammetric images, edges are mainly used as primitive features. Line geometry can be exploited to generate object structures (Heuvel, 1999; Hrabáček and Heuvel, 2000). In LIDAR data, a point cloud can be segmented into surface patches. Among the features, 3D surface patches are useful to generate building models (Jaynes et al., 1997). Surface patches can be grouped in relatively simple ways and used to generate 3D lines and 3D corners by intersecting adjacent surface patches (Ameri, 2000). Because of the strength of surface patches in building modelling, we use the surface patches as the fundamental primitives for modeling building shapes.
Regarding grouping mechanisms, we consider the efficiency of hypothesis generation process and the variation of roof surfaces. For efficient aggregation of surface patches into building parts, we suggest the use of a wing model. For grouping patches, adjacent relations among the patches are exploited in order to combine the patches into wing models. Wing models are also linked together to represent a complete building model.

A building obtained from building recognition is used to implement the suggested modeling method. The topological relations and the geometric accuracies of the modeled building entities are presented at the end of this chapter.

6.2 Segmentation of building roof surfaces

After recognizing building regions, roof surfaces can be segmented into planar surface patches where they are adjacent each other and can be merged. Segmentation process groups locally homogeneous regions (Brunn and Weidner, 1998; Vosselman and Dijkman, 2001; Lee and Schenk, 2002) and generates adjacency relations among the regions. Planar surface patches can be delineated interactively or automatically. We currently extract LIDAR points interactively by defining a closed polygon. In order to increase the quality of measurement, we exploit azimuth images. After the identification of points that belong to a planar surface patch, the surface parameters are estimated in a robust way.

6.2.1 Generation of azimuth images

In order to find the boundaries between surface patches, azimuth images are generated. The azimuth images represent the horizontal direction of local surface patches defined by a local window.
Because we use the direction of local patches for segmenting a roof surface, the parameters to be estimated are composed of two slope coefficients in the row and column directions. The parameters of the local surface patches are estimated based on a local coordinate system, where the horizontal origin is the center of window and the vertical origin is the mean of the pixel heights.

Hence, the mathematical model to estimate local surface parameters is written as:

\[
Z_{i,j} - \bar{Z} = S_r \cdot i + S_c \cdot j \tag{6.1}
\]

where

\[
i = \{-M, \ldots, 0, \ldots, M\}, j = \{-M, \ldots, 0, \ldots, M\}
\]

and

- \(M\) half of the local window size
- \(S_r, S_c\) surface parameters for local window area,
- \(Z_{i,j}\) height value at \((i, j)\) and
- \(\bar{Z}\) mean height of local pixels

We employ a Gauss-Markov model to estimate the surface parameters as:

\[
y = A \xi + e, \quad e \sim (0, \sigma_o^2 I_n) \tag{6.2}
\]

where

\[
y = \begin{bmatrix}
Z_{-M,-M} - \bar{Z} \\
\vdots \\
Z_{i,j} - \bar{Z} \\
\vdots \\
Z_{M,M} - \bar{Z}
\end{bmatrix}, \quad A = \begin{bmatrix}
-M & -M \\
\vdots & \vdots \\
i & j \\
\vdots & \vdots \\
M & M
\end{bmatrix} \quad \text{and} \quad \xi = \begin{bmatrix}
S_r \\
S_c
\end{bmatrix}.
\]

Because of the linearity of the model, the parameters and the residuals are directly computed as:

\[
\hat{\xi} = \begin{bmatrix}
\hat{S}_r \\
\hat{S}_c
\end{bmatrix} = (A^T A)^{-1} A^T y \tag{6.3}
\]
\[ \tilde{e} = y - A\tilde{\xi}. \] (6.4)

For robustness and efficiency in local parameter estimation, we apply a trimmed mean method. In order to estimate the parameters of a major surface, the method discards outlier pixels which have relatively high residuals. After fitting the pixel heights into a planar surface, the residuals from Eq. 6.4 are sorted. Then, inlier pixels are collected based on the relative rank of the residuals. Using the selected pixels, the parameters are estimated again by least square adjustment.

Using the estimated surface parameters - \( S_r \) and \( S_c \), normal vectors can be computed. Figure 6.1 illustrates the relation between the surface parameters and their corresponding normal vector. As can be seen, using the cross product of two vectors - \( \vec{a} \) and \( \vec{b} \), the relation can be represented as:

\[
\vec{n} = \vec{a} \times \vec{b} = \begin{bmatrix}
-\hat{S}_r \\
-\hat{S}_c \\
1
\end{bmatrix} = \begin{bmatrix}
n_r \\
n_c \\
n_Z
\end{bmatrix} \] (6.5)

Finally, the azimuth(\( \lambda \)) and the altitude(\( \phi \)) of a normal vector are computed from the elements of normal vector, \( \vec{n} \) as:

\[
\lambda = \tan^{-1} \left( \frac{n_c}{n_r} \right) = \tan^{-1} \left( \frac{-\hat{S}_c}{-\hat{S}_r} \right) \\
\phi = \tan^{-1} \left( \frac{1}{\sqrt{\hat{S}_r^2 + \hat{S}_c^2}} \right) \] (6.6)

Using the values in Eq. 6.6, azimuth and altitude images are generated. Among the images, azimuth images are exploited to distinguish crease edges in roof surfaces.
6.2.2 Generation of surface patches

Planar surface patches can be delineated interactively or automatically. We currently extract LIDAR points interactively by defining a closed polygon. After the identification of points that belong to a planar surface patch, the three plane parameters can be estimated. The relation between the plane parameters and the LIDAR points can be written as:

\[ Z_i = Z_o + S_X (X_i - \bar{X}) + S_Y (Y_i - \bar{Y}), \]  

(6.7)

where

- \( S_X, S_Y \) and \( Z_o \) surface parameters to be estimated
- \( X_i, Y_i \) coordinates of the centroid of the surface
- \( X_i, Y_i, Z_i \) coordinates of LIDAR points

From Eq. 6.7, the plane parameters are estimated by minimizing the perpendicular distance between LIDAR points and a plane. The perpendicular distance is computed
as:
\[
d_i = \frac{Z_i - [\hat{Z}_o + \hat{S}_X(X_i - \bar{X}) + \hat{S}_Y(Y_i - \bar{Y})]}{\sqrt{\hat{S}_X^2 + \hat{S}_Y^2 + 1}}.
\] (6.8)

For robust parameter estimation, a RANSAC based method is used to discard blunder points in a surface patch (Ameri, 2000). Three points are randomly selected among all the points. Then, three plane parameters in Eq. 6.7 are computed. Based on the plane defined by the three parameters, perpendicular distances in Eq. 6.8 are computed between all the points and the plane. The median of the distances is recorded as a property of the plane. These procedures are performed repeatedly for every randomly generated plane. Next, the optimal plane parameters are determined by finding a plane whose distance median is minimum among all the random planes. Now, outlier points are detected by comparing its distance from the plane to the standard accuracy of LIDAR data.

Finally, only the inlier points are used to estimate the plane parameters. A Gauss-Markov model is used to estimate the parameters and to provide the accuracy of the parameters as:
\[
y = \mathbf{A} \xi + \mathbf{e}, \quad \mathbf{e} \sim (\mathbf{0}, \sigma_o^2 \mathbf{I}_n) \tag{6.9}
\]

where
\[
y = \begin{bmatrix} d_1 \\ \vdots \\ d_i \\ \vdots \\ d_n \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \frac{\partial y}{\partial \Delta Z_o} & \frac{\partial y}{\partial \Delta S_X} & \frac{\partial y}{\partial \Delta S_Y} \end{bmatrix} \quad \text{and} \quad \xi = \begin{bmatrix} \Delta Z_o \\ \Delta S_r \\ \Delta S_c \end{bmatrix}.
\]

From the established Gauss-Markov model, the estimate of parameters and the prediction of residuals are acquired as:
\[
\hat{\xi} = \begin{bmatrix} \Delta Z_o \\ \Delta S_r \\ \Delta S_c \end{bmatrix} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \tag{6.10}
\]
\( \tilde{e} = y - A\hat{\xi}. \) \hspace{1cm} (6.11)

### 6.2.3 Generation of surface patch adjacency graphs

The adjacency relation between surface patches is important for several reasons, one being that adjacency is used to check if neighboring surface patches have similar parameters and thus can be merged into one patch. The adjacency relation can be established during the segmentation process.

If the segmentation is performed by region growing, seed pixels are selected within homogeneous regions, followed by growing regions until they are bounded by other surfaces. During this process adjacent patches are examined and merged if the fitting error does not exceed a tolerance. Missing information about the adjacency between surface patches can also be found by checking the surfaces that are close to each other.

The adjacency information between \( n \) surface patches is saved in a graph data structure called Surface Patch Adjacency Graph (SPAG). It is described by nodes and edges as:

\[
G_{SPAG} = \text{Graph}(N_{SPAG}; E_{SPAG}), \hspace{1cm} (6.12)
\]

where

\[
N_{SPAG} = \{SP_i, i = 1, \ldots, n\} \quad \text{and} \quad E_{SPAG} = \{SP_i \times SP_j, \text{Adjacent}(SP_i, SP_j) = True\}.
\]

This graph is undirected and the adjacent surface patches of a certain patch can be retrieved in bidirectional. Figure 6.2 illustrates a group of surface patches in a roof and their SPAG. The SPAG is exploited together with the geometric attributes of the surface patches to model buildings.
6.3 Generation of wing models

Grouping is a process of combining features into high level aggregations. It is one of the most important tasks in building hypothesis generation. The grouping mechanism needs to collect features efficiently and to represent the intermediate structures appropriately. Hence, one of the crucial factors in grouping process is the aggregation models.

In our approach, we suggest a wing model to aggregate features and represent generated building models. Figure 6.3 shows a processing flow for building modeling. As can be seen, the grouping mechanism is divided into several steps. First, antisymmetric planes are searched as strong clues to initiate wing models. Then, the remaining surface patches are examined if they can be grouped into the existing wing models from their adjacency graph SPAG. Finally, building models are generated by combining the generated wing models.
Figure 6.3: Building modelling procedure using wings
Figure 6.4: Anti-symmetry of two surface patches. The dotted lines between surface patches represent the adjacency relations. The pair of surface patches - $SP_i$ and $SP_j$ are used to initiate a wing model.

6.3.1 Initiation of wing models

The modeling process starts by computing the inner product of pairs of adjacent surfaces in order to find anti-symmetric surfaces. Anti-symmetry is a measure of how different the azimuths, $\lambda$, of surface patches are. Anti-symmetry is not related to the difference in altitude, $\phi$. Thus, the inner products are computed using only the azimuth angles for a pair of adjacent surface patches as illustrated in Figure 6.4.

The degree of anti-symmetry is computed using only horizontal components of the surface normal vector. As shown in Figure 6.4, the horizontal direction vector is
the unit vector projected from the normal vector and can be written as:

\[ \vec{n}_{hi} = \begin{bmatrix} \cos(\lambda_i) \\ \sin(\lambda_i) \end{bmatrix}, \quad (6.13) \]

where \( \lambda_i \) is the azimuth of the \( i \)th surface patch. Hence, the degree of anti-symmetry (\( \Lambda_{ij} \)) between a pair of surface patches - \( SP_i \) and \( SP_j \) can be defined as:

\[ \Lambda_{ij} = \Lambda(SP_i, SP_j) = -\text{InnerProduct}(\vec{n}_{hi}, \vec{n}_{hj}) = -\begin{bmatrix} \cos(\lambda_i) \\ \sin(\lambda_i) \end{bmatrix}^T \begin{bmatrix} \cos(\lambda_j) \\ \sin(\lambda_j) \end{bmatrix}. \quad (6.14) \]

From Eq. 6.14, a anti-symmetric value 1.0 represents the perfect oppositeness of two surfaces, while a value 0.0 implies the perfect orthogonality of them. In order to group only the reasonable pairs of the patches, we set the lower bound of the anti-symmetry as:

\[ \text{AntiSymmetric}(S_i, S_j) = \begin{cases} \text{True} & \text{if } \Lambda_{ij} > t \Lambda \\ \text{False} & \text{otherwise} \end{cases}. \quad (6.15) \]

The grouping process is performed repeatedly, starting with the surface pair of the highest anti-symmetry, \( \Lambda \), until the lower bound is met. Now, the wing models are intermediately defined as:

\[ \{W_k\} = \{SP_i \times SP_j\}, \quad \begin{cases} \text{Adjacent}(SP_i, SP_j) = \text{True} & \text{and} \\ \text{AntiSymmetric}(SP_i, SP_j) = \text{True} \end{cases} \quad (6.16) \]

where \( \{W_k\} \) is a set of the initiated wing models.

Figure 6.5 presents the wing models initiated from the surface patches in Figure 6.2. As can be seen, after initiation of wing models, the surface patches are also classified into two types - side surface patches of wing models and other surface patches which do not belong to any wing models.
6.3.2 Determination of ending surface patches of wing models

The initiated wing models are refined by adding information about their ending surfaces. A wing normally has two endings, each of which may have more than one surface patch. Possible ending types include gables, hips, and more complex structures such as cut-off hips as shown in Figure 6.6. Junctions are added as an ending type, computed as the intersection of a wing and certain adjacent surfaces.
Figure 6.7: Refined wing model after aggregating ending surfaces.

The ending type of wing model is determined by grouping the remaining surface patches which have not been classified into side planes of wing models. For example, in Figure 6.5, one remaining surface patch \( SP_2 \) is inspected to find the wing where the surface patch may belong. This is accomplished by using the adjacency relation from the graph \( SPAG \). If the remaining surface patch \( SP_r \) is adjacent to a surface patch \( SP_{W_i} \) that has been classified into one of the side planes of a wing model \( W_i \), then surface patch \( SP_r \) is grouped together into the wing model \( W_i \) as a surface at one of the endings. This relationship between a wing and a potential ending surface can be represented as:

\[
\text{EndingSurface}(SP_i, W_k) = \begin{cases} 
  \text{True} & \text{if } \exists SP_j, SP_j \in W_k; \\
  \text{Adjacent}(SP_i, SP_j) = \text{True} \\
  \text{False} & \text{otherwise}
\end{cases} \tag{6.17}
\]

Figure 6.7 shows two wing models generated from the initial surface patches. As can be seen, after aggregating ending surface patches, wing models contain information about their surfaces. This is needed for classifying building parts.
The final step of building modelling is the generation of a building hypothesis as a combination of wing models that one may consider as units of a complete building model. Figure 6.8 shows a building model at the final stage of modelling driven from the initial surface patches in Figure 6.2.
6.4 Boundary generation of building models

From the modeling process, a building model is represented as a composition of wings. However, prior to this process, the boundary lines of the model are not determined. In order to delineate the building shape, boundary lines have to be generated. Because boundary lines can be efficiently represented by corners, corner points are extracted by intersecting surface patches. This intersecting process requires at least three adjacent surface patches.

However, the corner points at eaves line usually have only two adjacent surfaces among the roof surface patches. To make up this discrepancy at the eaves corners, the building boundary contour hypothesized from the building recognition task is employed as an additional surface patch. Because most of corners along eaves lines are adjacent to the hypothesized contour, the contour plays a significant role to localize the corners of the building model.

We exploit the topological relations among the wings and the surface patches in order to generate the corner points automatically. At the ending parts of a wing, ridge corners are determined by intersecting two side surface patches and one ending patch. Eaves corners are also determined by intersecting one side surface patch, one ending patch and the surface driven from the building boundary contour. When the ending part of a wing is a junction, the ridge corners are determined by intersecting two side patches and the surface patch adjacent to the wing.

After the corner points are localized, they are linked together according to their topological relations in the wing. A ridge line is generated by connecting the ridge corners of the two ending parts. Eaves lines are also generated by connecting the eaves corners of the ending parts.
6.5 Experimental results of building model generation

In order to test the suggested modeling method, a relatively complex building is selected. The building is the H shape building located in block H. As shown in Figure 5.21, it seems difficult to generate a building model using optical features because there are many features which are not related to the physical building structures. Furthermore, optical edges for the surface boundaries are frequently missing. On the other hand, as shown in Figure 6.9(a), LIDAR data seem to provide good clues to extract the building surface structures.

However, the strategy to generate building models can be decided depending on the point density of LIDAR data. If the density is high enough to be segmented reliably, planar surface patches are promising features for building modeling. Because the point density of block H is relatively high, we apply our modeling method to building hypothesis generation.

Figure 6.9(a) presents an azimuth image generated by fitting local surfaces with a 5 by 5 filter - 2.5 m by 2.5 m in ground. In order to compute the parameters of the local major surface, a trimmed mean method is used. The figure looks like a shaded relief image, where light illuminates the LIDAR surface from the right side of the image. As can be seen, most of the building ridge lines are well distinguished. The figure demonstrates that the azimuth image is useful for LIDAR data segmentation automatically or interactively. Figure 6.9(b) shows the points collected interactively in order to segment the roof surface into surface patches.

After segmentation, the plane parameters of the surface patches are estimated in robust ways using a RANSAC approach. Table 6.1 shows the estimates of the surface parameters driven from the estimates of the plane parameters with accuracy 98
Figure 6.9: Corner points collected from the azimuth image
information. As can be seen, the parameters seem to be reasonable, when we consider
the nominal accuracy of LIDAR system is about 10 cm in the vertical direction. In
the table, it can be noticed that some of the surface patches - $SP_3, SP_5, SP_{10}$ and
$SP_{12}$ have relatively larger standard deviations. This is mainly caused by the small
number of data points within the patches.
<table>
<thead>
<tr>
<th>$SP_i$</th>
<th>No. points</th>
<th>No. inliers</th>
<th>$Z_o$ (m)</th>
<th>$\phi$ (deg)</th>
<th>$\lambda$ (deg)</th>
<th>$\hat{\sigma}_Z$ (cm)</th>
<th>$\hat{\sigma}_{Z_o}$ (cm)</th>
<th>$\hat{\sigma}_\phi$ (deg)</th>
<th>$\hat{\sigma}_\lambda$ (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SP_1$</td>
<td>321</td>
<td>198</td>
<td>-21.963</td>
<td>55.9221</td>
<td>67.3926</td>
<td>9.46</td>
<td>0.81</td>
<td>0.1420</td>
<td>0.0518</td>
</tr>
<tr>
<td>$SP_2$</td>
<td>80</td>
<td>48</td>
<td>-21.381</td>
<td>56.6296</td>
<td>-113.0041</td>
<td>9.95</td>
<td>1.72</td>
<td>0.3305</td>
<td>0.3218</td>
</tr>
<tr>
<td>$SP_3$</td>
<td>30</td>
<td>17</td>
<td>-22.090</td>
<td>50.5581</td>
<td>157.3486</td>
<td>10.39</td>
<td>3.26</td>
<td>1.0332</td>
<td>1.3391</td>
</tr>
<tr>
<td>$SP_4$</td>
<td>163</td>
<td>111</td>
<td>-21.451</td>
<td>55.7576</td>
<td>-111.3309</td>
<td>8.78</td>
<td>1.01</td>
<td>0.1961</td>
<td>0.1342</td>
</tr>
<tr>
<td>$SP_5$</td>
<td>21</td>
<td>8</td>
<td>-21.974</td>
<td>58.2290</td>
<td>-24.4934</td>
<td>7.95</td>
<td>3.31</td>
<td>0.6578</td>
<td>1.6864</td>
</tr>
<tr>
<td>$SP_6$</td>
<td>123</td>
<td>71</td>
<td>-21.466</td>
<td>57.9406</td>
<td>158.7935</td>
<td>9.38</td>
<td>1.31</td>
<td>0.1991</td>
<td>0.2337</td>
</tr>
<tr>
<td>$SP_7$</td>
<td>115</td>
<td>59</td>
<td>-21.850</td>
<td>59.6287</td>
<td>-21.6234</td>
<td>9.78</td>
<td>1.48</td>
<td>0.2175</td>
<td>0.2859</td>
</tr>
<tr>
<td>$SP_8$</td>
<td>259</td>
<td>164</td>
<td>-21.533</td>
<td>58.7602</td>
<td>-112.2475</td>
<td>10.60</td>
<td>0.97</td>
<td>0.1486</td>
<td>0.0802</td>
</tr>
<tr>
<td>$SP_9$</td>
<td>88</td>
<td>56</td>
<td>-21.679</td>
<td>58.0306</td>
<td>67.4544</td>
<td>10.26</td>
<td>1.62</td>
<td>0.2762</td>
<td>0.3862</td>
</tr>
<tr>
<td>$SP_{10}$</td>
<td>36</td>
<td>20</td>
<td>-21.872</td>
<td>61.5469</td>
<td>155.0470</td>
<td>9.26</td>
<td>2.35</td>
<td>0.3654</td>
<td>1.0086</td>
</tr>
<tr>
<td>$SP_{11}$</td>
<td>199</td>
<td>106</td>
<td>-21.970</td>
<td>59.5001</td>
<td>68.6793</td>
<td>9.19</td>
<td>1.04</td>
<td>0.1564</td>
<td>0.2017</td>
</tr>
<tr>
<td>$SP_{12}$</td>
<td>22</td>
<td>10</td>
<td>-22.554</td>
<td>63.2258</td>
<td>-28.1371</td>
<td>9.35</td>
<td>3.31</td>
<td>0.8592</td>
<td>1.8710</td>
</tr>
</tbody>
</table>

Table 6.1: Estimates of surface patch parameters

<table>
<thead>
<tr>
<th>$SP_i$</th>
<th>No. points</th>
<th>No. inliers</th>
<th>$Z_o$ (m)</th>
<th>$\phi$ (deg)</th>
<th>$\lambda$ (deg)</th>
<th>$\hat{\sigma}_Z$ (cm)</th>
<th>$\hat{\sigma}_{Z_o}$ (cm)</th>
<th>$\hat{\sigma}_\phi$ (deg)</th>
<th>$\hat{\sigma}_\lambda$ (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SP_{13}$</td>
<td>243</td>
<td>158</td>
<td>-21.474</td>
<td>55.8203</td>
<td>-112.4525</td>
<td>9.68</td>
<td>0.93</td>
<td>0.1749</td>
<td>0.0588</td>
</tr>
<tr>
<td>$SP_{14}$</td>
<td>287</td>
<td>167</td>
<td>-21.824</td>
<td>59.7142</td>
<td>67.0184</td>
<td>9.88</td>
<td>0.89</td>
<td>0.1347</td>
<td>0.0711</td>
</tr>
</tbody>
</table>

Table 6.2: Estimates of the merged surface patch parameters
Using the plane parameters, the mutually consistent surface patches are merged into one planar patch. Table 6.2 shows the surface parameters after merging. The surface patch $SP_{13}$ is generated by merging $SP_2$ and $SP_4$ and the surface patch $SP_{14}$ generated by merging $SP_9$ and $SP_{11}$ in Table 6.2.

Figure 6.10 illustrates the surface patches and their initial adjacency relations. This information is recorded into a graph data structure SPAG and used for aggregating surface patches into wing models. As can be seen, the boundaries among the surface patches are not delineated yet. Regarding modeling, hence, the boundaries are not generated until the grouping process is complete.

In order to group the surface patches into wing models, the degrees of anti-symmetry between all the adjacent patches were computed. Table 6.3 shows the resulting degrees of anti-symmetry. As can be seen, the values can be immediately divided into two groups. One group of values are close to 1.0 and the others are close to 0.0. For the lower bound of the anti-symmetry value, 0.9 was set. This corresponds to the angle difference from $154^\circ$ to $206^\circ$. Now, wing models are generated using the surface patches which belong to the first group.

Table 6.4 summarizes the properties of the produced wing models. After generating wing models, the remaining patches are checked to determine whether they can be one of the ending surface patches of a wing model. Then, as shown in Table 6.4, the type of building endings are determined. The building is now described by three adjacent wing models whose ending types are classified as hip and junction.

Table 6.5 shows the properties of the corners generated by intersecting the adjacent surface patches. The adjacent patches are determined using the topological
Figure 6.10: Surface patch adjacency graph of the building H
\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Link & \(SP_i\) & \(SP_j\) & \(\lambda_i\) (deg) & \(\lambda_j\) (deg) & \(\Lambda_k\) \\
\hline
Link_1 & \(SP_1\) & \(SP_3\) & 67.4813 & 159.7085 & 0.04 \\
Link_2 & \(SP_1\) & \(SP_3\) & 67.4813 & -27.9731 & 0.10 \\
Link_3 & \(SP_1\) & \(SP_{13}\) & 67.4813 & -112.2569 & 1.00 \\
Link_4 & \(SP_3\) & \(SP_{13}\) & 159.7085 & -112.2569 & -0.03 \\
Link_5 & \(SP_5\) & \(SP_{13}\) & -27.9731 & -112.2569 & -0.10 \\
Link_6 & \(SP_6\) & \(SP_7\) & 158.9182 & -21.6839 & 1.00 \\
Link_7 & \(SP_6\) & \(SP_{13}\) & 158.9182 & -112.2569 & -0.02 \\
Link_8 & \(SP_6\) & \(SP_{14}\) & 158.9182 & 66.8883 & 0.04 \\
Link_9 & \(SP_7\) & \(SP_{13}\) & -21.6839 & -112.2569 & 0.01 \\
Link_10 & \(SP_7\) & \(SP_{14}\) & -21.6839 & 66.8883 & -0.02 \\
Link_11 & \(SP_8\) & \(SP_{10}\) & -112.1941 & 158.7527 & -0.02 \\
Link_12 & \(SP_8\) & \(SP_{12}\) & -112.1941 & -28.3936 & -0.11 \\
Link_13 & \(SP_8\) & \(SP_{14}\) & -112.1941 & 66.8883 & 1.00 \\
Link_14 & \(SP_{10}\) & \(SP_{14}\) & 158.7527 & 66.8883 & 0.03 \\
Link_15 & \(SP_{12}\) & \(SP_{14}\) & -28.3936 & 66.8883 & 0.09 \\
\hline
\end{tabular}
\caption{Anti-symmetry(\(\Lambda_k\)) between the adjacent surface patches}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Wing & \(SS_1\) & \(SS_2\) & \(ES_1\) & \(ES_2\) & Class of \(ES_1\) & Class of \(ES_2\) \\
\hline
\(W_1\) & \{\(SP_1\)\} & \{\(SP_{13}\)\} & \{\(SP_3\)\} & \{\(SP_3\)\} & hip & hip \\
\(W_2\) & \{\(SP_6\)\} & \{\(SP_7\)\} & \{\(SP_{13}\)\} & \{\(SP_{14}\)\} & junction & junction \\
\(W_3\) & \{\(SP_8\)\} & \{\(SP_{14}\)\} & \{\(SP_{10}\)\} & \{\(SP_{12}\)\} & hip & hip \\
\hline
\end{tabular}
\caption{Description of the generated wing models}
\end{table}

104
relations described in Table 6.3 and Table 6.4. Eighteen corners are localized by intersecting the hypothesized building contour and the ten planar surface patches in the wing models. As can be seen, all corners could be localized by intersecting three adjacent surface patches. However, the accuracy of the corners is considerably different depending on their types in both horizontal and vertical directions. The accuracy of the surface from building recognition is set to 1 meter because raw contours are generated at 0.5 m of vertical interval.

Finally, Figure 6.12 shows the generated building model together with the raw LIDAR surface presented in Figure 6.11. As can be seen, two surfaces are fairly consistent in both the vertical and horizontal directions. However, the eaves lines of the building model tend to be slightly shifted toward the inside of the building region. This slight shift of the eaves lines will be resolved by integrating the features of the aerial images in model verification.
<table>
<thead>
<tr>
<th>C3D</th>
<th>Surface patches ( {SP_i} )</th>
<th>Coordinates</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( X ) (m)</td>
<td>( Y ) (m)</td>
</tr>
<tr>
<td>C3D1</td>
<td>{SP_1, SP_3, SP_{13}}</td>
<td>117.164</td>
<td>80.793</td>
</tr>
<tr>
<td>C3D2</td>
<td>{SP_5, SP_{13}, SP_{15}}</td>
<td>119.655</td>
<td>74.240</td>
</tr>
<tr>
<td>C3D3</td>
<td>{SP_1, SP_5, SP_{15}}</td>
<td>124.504</td>
<td>83.369</td>
</tr>
<tr>
<td>C3D4</td>
<td>{SP_1, SP_3, SP_{13}}</td>
<td>79.804</td>
<td>96.182</td>
</tr>
<tr>
<td>C3D5</td>
<td>{SP_1, SP_3, SP_{15}}</td>
<td>75.445</td>
<td>103.709</td>
</tr>
<tr>
<td>C3D6</td>
<td>{SP_3, SP_{13}, SP_{15}}</td>
<td>71.792</td>
<td>93.828</td>
</tr>
<tr>
<td>C3D7</td>
<td>{SP_3, SP_6, SP_7}</td>
<td>92.953</td>
<td>90.906</td>
</tr>
<tr>
<td>C3D8</td>
<td>{SP_7, SP_{13}, SP_{15}}</td>
<td>96.415</td>
<td>83.750</td>
</tr>
<tr>
<td>C3D9</td>
<td>{SP_6, SP_{13}, SP_{15}}</td>
<td>85.863</td>
<td>88.069</td>
</tr>
<tr>
<td>C3D10</td>
<td>{SP_6, SP_7, SP_{14}}</td>
<td>84.215</td>
<td>68.581</td>
</tr>
<tr>
<td>C3D11</td>
<td>{SP_{14}, SP_7, SP_{15}}</td>
<td>91.675</td>
<td>71.830</td>
</tr>
<tr>
<td>C3D12</td>
<td>{SP_{6}, SP_{14}, SP_{15}}</td>
<td>81.308</td>
<td>76.254</td>
</tr>
<tr>
<td>C3D13</td>
<td>{SP_6, SP_{12}, SP_{14}}</td>
<td>106.088</td>
<td>59.464</td>
</tr>
<tr>
<td>C3D14</td>
<td>{SP_{6}, SP_{12}, SP_{15}}</td>
<td>108.151</td>
<td>51.927</td>
</tr>
<tr>
<td>C3D15</td>
<td>{SP_{12}, SP_{14}, SP_{15}}</td>
<td>113.804</td>
<td>62.386</td>
</tr>
<tr>
<td>C3D16</td>
<td>{SP_8, SP_{14}, SP_{10}}</td>
<td>71.009</td>
<td>74.118</td>
</tr>
<tr>
<td>C3D17</td>
<td>{SP_8, SP_{10}, SP_{15}}</td>
<td>63.100</td>
<td>70.306</td>
</tr>
<tr>
<td>C3D18</td>
<td>{SP_{10}, SP_{14}, SP_{15}}</td>
<td>67.675</td>
<td>82.073</td>
</tr>
</tbody>
</table>

Table 6.5: Building corners generated by intersecting surface patches
Figure 6.11: Raw LIDAR surface

Figure 6.12: Generated building model in LIDAR surface
CHAPTER 7

HYPOTHESIS VERIFICATION OF BUILDING MODELS

7.1 Introduction

The generated hypotheses are now verified with features and domain knowledge. To make the verification process as robust and as independent from the hypotheses generation as possible we integrate extracted features from aerial images and LIDAR data. The verification process is combined with estimating the parameters of the object models. The generated hypotheses are considered approximations of real objects with respect to geometry and topology.

From aerial images, extracted features are used for evaluating the correspondence between the models back projected into images and the features in image domain. During back projection, the geometry and topology of the generated three dimensional models are transformed into two dimensional hypothesized features and the topological relations between them. The sensor model is also considered.

From LIDAR data, the extracted surface patches on roof regions are incorporated for checking the validity of the generated models. Information concerning boundary contours, determined from the building recognition process, is very useful to check the validity and to reduce the search space when estimating the model parameters.
7.2 Verification with features from aerial images

The verification of building models using aerial images is accomplished by comparing the back projected models with extracted observed features such as edges, corners and regions. For extracting edges we use the Canny edge detector (Canny, 1986), followed by linking edge pixels to edge entities.

The verification process determines the difference between the boundaries back projected from the building models and the edges in the images (Ameri, 2000; Rottensteiner, 2001). Figure 7.1 shows the perpendicular distance from a edge point $p_k(x_k, y_k)$ to the boundary line. Here, the boundary line in aerial images is represented by the two corner points - $p_i(x_i, y_i)$ and $p_j(x_j, y_j)$. We establish the estimation model between the back projected corners and the extracted edges directly, without estimating the line parameters of the model in the image domain, by introducing connectivity of corners, which can be written as:

$$f_{dk} = \frac{(y_j - y_i)x_k - (x_j - x_i)y_k - (y_j - y_i)x_i + (x_j - x_i)y_i}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} + e_{dk},$$

(7.1)

where $(x_i, y_i)$ and $(x_j, y_j)$ are the coordinates of back projected corners of models which should be estimated and $(x_k, y_k)$ are the coordinates of extracted edge locations.

The sensor model of aerial frame images is defined by the collinearity condition that defines the relation between the coordinates of three dimensional object model and the corresponding model in images as shown in Figure 7.2.
Figure 7.1: Distance to be minimized between edge points and a boundary line

Figure 7.2: Collinearity constraint from the sensor model of the aerial images
The stochastic condition can be written as:

\[
\begin{align*}
  f_{xi} &= x_i - x_p + f_r^1(X_i - X_o) + r_{21}(Y_i - Y_o) + r_{31}(Z_i - Z_o) + e_{xi} \\
  f_{yi} &= y_i - y_p + f_r^2(X_i - X_o) + r_{22}(Y_i - Y_o) + r_{32}(Z_i - Z_o) + e_{yi}
\end{align*}
\]  

(7.2)

7.3 Verification with features from LIDAR data

Surface patches composing the hypothesized building model are used as stochastic resources providing surface parameter information. The condition can be written as:

\[
  f_{Sp}^m = Z_i - \hat{Z}_o^m - \hat{S}_X^m (X_i - \bar{X}_m) - \hat{S}_Y^m (Y_i - \bar{Y}_m) + e_{Sp}^m
\]

(7.3)

It is complicated to remove blunder edges which are parallel to the boundary edges of building model. We can use the height information of boundary contours from building recognition to discard these blunder edges during iteration of parameter estimation. The stochastic model of the contour information can be used for the corners along eaves line of building model as:

\[
  f_{He} = Z_i - H_e + e_{He}
\]

(7.4)
7.4 Experimental results of building model verification

In order to verify the generated building model, a pair of subset aerial images from the full images are used together with their orientation information. The orientation information consists of the estimate of exterior orientation parameters from the coregistration process and the interior orientation information driven from the affine transformation coefficients for the full images. Figure 7.3 shows the images which are scanned in 24 $\mu$m, where a pixel corresponds to about 9 cm by 9 cm in ground.

Figure 7.4 shows the generated building model back projected in the aerial images. As can be seen, the ridge lines of the building model correspond closely to the edges of ridge lines in aerial images. However, some discrepancies are also found along the eaves lines between the model and the gray contrasts in aerial images. As summarized in Table 6.5, these discrepancies are already expected from the large standard deviation of corner points at eaves lines in the generated model. This problem originates from the low number of points at the ending surface patches and from the approximated height from the hypothesized building boundary contour. Nevertheless, the back projected wire frame provides a remarkably reduced search space to find the corresponding features in image domain.
Figure 7.3: Aerial images for block H
Figure 7.4: A generated building model back projected in the aerial images.
Model verification can be performed based on corner and line features. Corner features can be directly extracted by using local filtering methods (Deriche and Giraudon, 1993; Förstner, 1994). In this study, however, in order to verify the model based on the boundary lines, straight lines are exploited. Figure 7.5 shows the straight lines from the edge detection and linking processes. Edge detection was performed using Canny edge operator with sigma value 2.0 and its corresponding filter size, 11 by 11. We applied a binomial operator to reduce processing time. After suppressing non-maxima edge pixels, the remaining pixels are linked into connected edges. During the edge linking process, geometric straightness at the junction of edges is also considered to extract long straight lines. High and low threshold values are chosen from the histogram of edge strengths at 70 % and 30 %, respectively. In addition, short edge lines are discarded. Finally, edge lines are segmented into straight lines with iterative splitting and merging processes.

As shown in Figure 7.5, straight lines are properly detected throughout the image. However, there are still many missing parts of surface boundary lines, in particular in the building roofs, due to the low contrast. In the regions of other buildings, most ridge lines are missing. As can be seen, crease edges at the intersection of the side surfaces and the ending surfaces are not detected well. However, by introducing the ridge lines from the generated building model and topological information between features, this problem is considerably resolved.
Figure 7.5: Straight lines extracted from aerial images for block H
Figure 7.6 presents the selected edges using buffer regions. To make search spaces for edges, buffer regions are generated based on the back projected lines of the model. The size of buffer is set by considering the resulting standard deviations of the original model summarized in Table 6.5. Here, we use 22 pixel wide buffers which corresponds to about two meter wide elongated areas in ground. As can be seen, the corresponding straight lines are retrieved by checking the edges in the overlapped regions of straight lines and the buffer regions. After obtaining a list of straight lines, some of the straight lines are discarded if they are not plausible by checking the radiometric consistency with image regions and by evaluating the geometric residuals during adjustment process.
Figure 7.6: Edges retrieved in building boundary buffers
The building models are verified by evaluating the consistency of topology and geometry. For checking the topology of the model, connectivity and coplanarity of corners are considered in adjustment equations. The geometric consistency is evaluated by the difference of the hypothesized corners and the estimated corners. As initial locations of corners, the hypothesized model corners are used.

Figure 7.7 demonstrates that the corners are moved toward corresponding points in aerial images by integrating edges in the adjustment model.

Because a pair of aerial images are a popular data set for cartographic map compilation and object reconstruction of the real world, it may be meaningful to check the geometric accuracy of reconstructed buildings using only the features extracted from aerial images. In order to implement this test, only the straight lines from aerial images are used to estimate the coordinates of the corners. This can be considered as a complex type of space intersection that combines sensor models of the images and the topological relations between image features together.

Table 7.1 summarizes the locations and the accuracies of the estimated corners using only aerial features. As can be seen, the resulting accuracies are reasonable even though the accuracies of the orientation parameters and the correlation matrices are different from the accuracies of the usual aerial image triangulation. Compared to the accuracies of corners estimated by LIDAR surface patches in Table 6.5, the accuracies of corners by aerial images are evenly estimated without depending on their types such as ridge and eaves. However, the vertical accuracies are at least two times lower than their horizontal accuracies. This is mainly caused by relatively weak geometry of the pair of aerial images in vertical direction.
Figure 7.7: Corners approximated and estimated through the verification process. Initial approximation and estimation (upper pair) and final result (lower pair). The figures show the approximated (green) and estimated (red) corners at each iteration.
<table>
<thead>
<tr>
<th>C3D</th>
<th>Type</th>
<th>Coordinates</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X (m)</td>
<td>Y (m)</td>
</tr>
<tr>
<td>C3D1</td>
<td>ridge</td>
<td>115.985</td>
<td>81.319</td>
</tr>
<tr>
<td>C3D2</td>
<td>eaves</td>
<td>119.343</td>
<td>74.007</td>
</tr>
<tr>
<td>C3D3</td>
<td>eaves</td>
<td>123.660</td>
<td>84.352</td>
</tr>
<tr>
<td>C3D4</td>
<td>ridge</td>
<td>78.947</td>
<td>96.684</td>
</tr>
<tr>
<td>C3D5</td>
<td>eaves</td>
<td>75.997</td>
<td>103.698</td>
</tr>
<tr>
<td>C3D6</td>
<td>eaves</td>
<td>71.931</td>
<td>93.164</td>
</tr>
<tr>
<td>C3D7</td>
<td>ridge</td>
<td>92.883</td>
<td>90.698</td>
</tr>
<tr>
<td>C3D8</td>
<td>eaves</td>
<td>96.899</td>
<td>82.829</td>
</tr>
<tr>
<td>C3D9</td>
<td>eaves</td>
<td>85.383</td>
<td>87.847</td>
</tr>
<tr>
<td>C3D10</td>
<td>ridge</td>
<td>84.420</td>
<td>67.665</td>
</tr>
<tr>
<td>C3D11</td>
<td>eaves</td>
<td>92.265</td>
<td>71.857</td>
</tr>
<tr>
<td>C3D12</td>
<td>eaves</td>
<td>80.865</td>
<td>76.558</td>
</tr>
<tr>
<td>C3D13</td>
<td>ridge</td>
<td>105.554</td>
<td>59.942</td>
</tr>
<tr>
<td>C3D15</td>
<td>eaves</td>
<td>113.826</td>
<td>63.226</td>
</tr>
<tr>
<td>C3D16</td>
<td>ridge</td>
<td>70.917</td>
<td>73.989</td>
</tr>
<tr>
<td>C3D17</td>
<td>eaves</td>
<td>62.208</td>
<td>70.348</td>
</tr>
<tr>
<td>C3D18</td>
<td>eaves</td>
<td>67.578</td>
<td>82.027</td>
</tr>
</tbody>
</table>

Table 7.1: Estimates of corners using features from aerial images

Figure 7.8 demonstrates the discrepancies between two surfaces from LIDAR data and aerial images. As can be seen, there are obvious differences in their heights. This is mainly due to the low accuracies of aerial images in the vertical direction as summarized in Table  7.1.

Finally, we verified the building model using the features from both aerial images and LIDAR data. Table  7.2 shows the location and the accuracies of the estimated corners by integrating the geometry and topology of all the generated features from two sensors. As can be seen, vertical accuracies are highly improved when compared to the case of using only aerial image features. However, the horizontal accuracies
Figure 7.8: Reconstructed building model using aerial image features only.
<table>
<thead>
<tr>
<th>C3D</th>
<th>Type</th>
<th>Coordinates</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X (m)</td>
<td>Y (m)</td>
</tr>
<tr>
<td>C3D1</td>
<td>ridge</td>
<td>116.858</td>
<td>80.936</td>
</tr>
<tr>
<td>C3D2</td>
<td>eaves</td>
<td>119.430</td>
<td>74.017</td>
</tr>
<tr>
<td>C3D3</td>
<td>eaves</td>
<td>123.880</td>
<td>83.895</td>
</tr>
<tr>
<td>C3D4</td>
<td>ridge</td>
<td>79.530</td>
<td>96.306</td>
</tr>
<tr>
<td>C3D5</td>
<td>eaves</td>
<td>75.789</td>
<td>103.651</td>
</tr>
<tr>
<td>C3D6</td>
<td>eaves</td>
<td>71.830</td>
<td>93.553</td>
</tr>
<tr>
<td>C3D7</td>
<td>ridge</td>
<td>93.035</td>
<td>90.832</td>
</tr>
<tr>
<td>C3D8</td>
<td>eaves</td>
<td>96.851</td>
<td>82.872</td>
</tr>
<tr>
<td>C3D9</td>
<td>eaves</td>
<td>85.363</td>
<td>87.865</td>
</tr>
<tr>
<td>C3D10</td>
<td>ridge</td>
<td>84.278</td>
<td>68.484</td>
</tr>
<tr>
<td>C3D11</td>
<td>eaves</td>
<td>92.255</td>
<td>72.062</td>
</tr>
<tr>
<td>C3D12</td>
<td>eaves</td>
<td>81.094</td>
<td>76.658</td>
</tr>
<tr>
<td>C3D13</td>
<td>ridge</td>
<td>105.599</td>
<td>59.706</td>
</tr>
<tr>
<td>C3D14</td>
<td>eaves</td>
<td>108.868</td>
<td>51.605</td>
</tr>
<tr>
<td>C3D15</td>
<td>eaves</td>
<td>113.940</td>
<td>63.009</td>
</tr>
<tr>
<td>C3D16</td>
<td>ridge</td>
<td>70.967</td>
<td>74.126</td>
</tr>
<tr>
<td>C3D17</td>
<td>eaves</td>
<td>62.372</td>
<td>70.290</td>
</tr>
<tr>
<td>C3D18</td>
<td>eaves</td>
<td>67.645</td>
<td>81.959</td>
</tr>
</tbody>
</table>

Table 7.2: Estimates of corners by integrating LIDAR data and aerial images

are decreased. This seems to be caused by distributing error sources evenly to all directions.

Figure 7.9 shows the back projected lines of the verified model. Table 7.2 shows final estimates of the corner points. As can be seen, the model is not only verified in aerial images but its accuracy is also well distributed by integrating aerial image features. Figure 7.10 illustrates the building reconstructed by our suggested fusion method in the raw laser surface. As can be seen, there are tight correspondences in vertical direction and wall outlines seem to be well localized.
Figure 7.9: A verified building model back projected in aerial images.
Figure 7.10: Reconstructed building by integrating the features from both aerial images and LIDAR data
CHAPTER 8

CONCLUSIONS

We presented new methods for automatic building recognition and reconstruction. Aerial images and LIDAR data were integrated by fusing features and sensor models. A rather complex building model was modeled successfully by aggregating surface patches into wing models. The generated models were verified and refined by combining the extracted features in a stochastic way.

While traditional methods perform registration using control points, we performed co-registration based on control lines. Important components were considered such as the common reference frame, control features and mathematical relations between the features from the two different sensors. Regarding the common reference frame, because LIDAR data provide object positions directly, LIDAR surfaces are set as the reference frame. However, because raw LIDAR data do not provide precisely localized features in direct ways, the control features are extracted after segmenting raw data into surface patches. Using the control lines from LIDAR data, we oriented the aerial images. Although the orientation parameters show high correlations due to the limited number of control features, the results of the localized corners using only the features from the oriented aerial images present reasonable accuracies of the object coordinates.
With regard to building recognition, we proposed a method to recognize building boundaries not only in the horizontal direction but also in the vertical direction. We exploited contours as the promising features to recognize buildings. Contour features were considered as the representative features to extract vertical discontinuities. While local filtering methods give the local response in a certain window and region growing methods usually evaluate consistency in local regions, contour based methods evaluate global properties of surfaces. Hence, when compared with other local filtering methods or general region growing methods, the contour based method detects building boundaries reliably.

We modeled the adjacency relations between contours using a hierarchical tree-like graph data structure, a contour graph. The contour graphs were efficiently established using the ray crossing algorithm which tests the relation point-in-polygon. Vertical discontinuity is assessed based on the slopes between contours. In order to compute the slopes precisely, the contours were extracted in vector format. Vector format contours enable the extraction of multiple contours from a single pixel and the localization of the contours in sub-pixel. Hence, the slopes can be computed reliably using the vector format contours.

In order to detect vertical discontinuity among surfaces, we exploited domain knowledge about slope variations in buildings and background objects. The results show that contours can be classified by introducing reasonable slope criteria from building domain knowledge.

We implemented our building recognition approach using two data sets. The two data sets have several opposite characteristics. One is a low point density data set and the other a relatively high point density data set. The first data set includes
a residential urban block; the second set includes a commercial urban block. The results using low density data show that our approach is robust to the variation of the point density in building recognition. In addition, although the detected building boundaries do not closely localize the actual building boundaries, the heights of the boundary contours and the LIDAR points within the contours can provide good clues for finding the features relevant to the hypothesized buildings. We also suggest change detection using discrepancies between aerial image features and LIDAR data features.

We designed a method to estimate the plane parameters of surface patches in a robust way and to remove datum dependency in parameter estimation. The surface patches are extracted using boundary contours and interactively measured points. We exploit a RANSAC method to estimate the plane parameters robustly. In order to remove the datum dependency in parameter estimation, the origin of the surface patches is moved to the centroid of the surface patches. For accurate estimation in both the horizontal and vertical directions, the perpendicular distances between a model plane and the LIDAR points are used for adjustment.

We proposed a new method for building modeling. A wing model is suggested as the core model for grouping 3D surface patches into building parts. Both topological and geometric properties of surface patches are utilized together to initiate wing models. The anti-symmetric property between two adjacent surface patches is exploited as a grouping cue to initiate wing models. The results indicate that building structures can be fairly well grouped by the anti-symmetric property of surface patches.

Based on the surface patch adjacency graph, the initiated wing models are refined by adding ending surface patches. We retrieved topological relations between building
structures at multiple levels in order to find the ending surface patches. Hierarchical adjacency relations between the generated wing models and the remaining surface patches are used for generating junction surfaces and for completing the wing models. In the experiment, the refined wing models corresponded properly to the actual building parts. A complete building model was generated by combining wing models. As shown in the procedure, the grouping mechanism is fairly flexible to accommodate various building and part models.

Boundary delineation of the building model was performed after generating a complete building model. The boundaries of surface patches were extracted by intersecting adjacent surface patches. Here, we solved a crucial problem in boundary delineation. The problem is that eaves corners cannot be localized because of the limited number of surface patches adjacent to them. In order to solve this problem, we exploited the surface of the recognized building contour as an additional surface patch and localized the eaves corners by intersecting two roof surface patches and the contour surface.

The results show that 3D surface patches are highly useful features to generate building models. In addition, our method is shown to be an efficient and flexible method for grouping 3D surface patches.

In building model verification, we integrated the building model generated from LIDAR data with features extracted from aerial images. The correspondences between the 3D building model and the 2D image features are verified through the sensor model established from co-registration. We established direct relations between the object corners and the features in aerial images. The topological relations defined in the building model, such as connectivity, are established in the adjustment
model. Within the boundary buffers, edge lists rather than individual edge pixels are exploited. This method has the advantage of removing invalid edges based on the properties that belong to the global edge structures. Our method reduced the time required to retrieve relevant edges and enhanced the capacity of blunder edge detection in the verification task.

Based on the proposed methods and the experimental results, we summarize some important factors in order to extend our research as follows:

- In automatic building reconstruction, building modeling is considered as one of the most crucial tasks. The modeling process can be extended by integrating the features from both aerial images and LIDAR data. Graphical modeling can also consider the probabilistic descriptions about domain knowledge Buntine (1994).

- We suggest the utilization of corner models, wing models and face models together to group the available features more flexibly. This is particularly important when the data sets do not provide enough features for generating a certain type of grouping model.

- The recognized building boundaries can be refined by analyzing LIDAR surfaces further or by comparing them with features in aerial images.

- For realistic scene description of buildings and cities, color aerial images can be useful.

- Other types of sensor data, in particular, multispectral images and hyperspectral images, can be integrated together to increase the level of description of
the object space. Using the spectral information associated with the objects, such as buildings, trees and vegetation areas objects can be identified more thoroughly.

- Sensor accuracy can be assessed by comparing corresponding features from different sensor data (Schenk et al., 2001) and used for sensor calibration.


136


Rockett, P., ???? The accuracy of sub-pixel localisation in the canny edge detector. URL citeseer.nj.nec.com/rockett99accuracy.html


