INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
SURFACE RECONSTRUCTION USING 3D LINEAR FEATURES

DISSERTATION

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

By
Deeboon Methakullachat, B.Eng., M.Sc.

The Ohio State University
2002

Dissertation Committee:
Prof. Anton F. Schenk, Adviser
Prof. Ayman F. Habib
Dr. Beata Csatho

Approved by
Adviser
Graduate Program in
Geodetic Science and
Surveying

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Automatic surface reconstruction has been an important research topic in digital photogrammetry for many years. Its central part is to solve the matching problem automatically. Most algorithms used to solve the matching problem can be categorized as either area-base techniques or feature-based techniques. Over the past years, feature-based matching has gained more and more popularity for it is the method that supports activities in object recognition and image understanding. Hence it fits well with the new definition of modern digital photogrammetry.

Recent research in digital photogrammetry and computer vision demonstrate the success of using linear features for solving the matching problem, which results in data sets of 3D linear features in the object space. While more research in linear feature matching is expected, there are only a few investigations regarding the construction of 3D linear features for surface reconstruction. There is clearly a lack of research in reconstructing surfaces by using 3D linear features in the object space.

This research is concerned with the reconstruction of surfaces from 3D linear features, in particular 3D lines, in the object space. Intermediate data processing will
include 3D line segmentation, 3D line grouping, 3D line linking, classification into ground and non-ground lines, plane finding and plane fitting. Planes are checked whether they are real physical planes. If this is the case the boundary of the plane are determined.

An abstract representation of surfaces by planes, including two types of plane formats, is proposed. The advantages over traditional surface representation, DEM and TIN, are discussed. It is expected that the concept of surface construction using 3D lines in this research may be as well applicable to object recognition, in terms of moving from a classical search in the 2D image space to a new 3D domain in the object space.

The recommended surface using 3D lines will facilitate subsequent vision tasks such as object recognition. Spatial reasoning in object space is much more powerful than reasoning in 2D image space.
Dedicated to photogrammetrists, surveyors, geodesists, cartographers, and all people who involve in mapping
ACKNOWLEDGMENT

• I thank the Faculty of Engineering of Kasetsart University who gave me a scholarship to study at OSU in such a hard time when Thailand economy was in jeopardy.

• I thank all the faculty members at the Department of Civil Engineering, Kasetsart University, especially the department head, Dr. Pornsak Pudhaphongsiriporn. Dr. Wiroj Rujopakarn, Dr. Korchoke Chantawarangul, Prof. Araya Nualindha, and others have helped extend my funding in the last year.

• I thank Prof. Gomet Hongpakdee who was passed away in January 1999. He is my teacher, my boss, my big brother and my best friend.

• I thank my Engineering (64) friends at Chulalongkom University in Thailand for their encouragement, mental support and financial support. I am so lucky to have such good friends. They are Pong+, Chit, Keng, Sutat, Man and Tern, Tia, Poo, Kom, Gong, Bird, Arj, Kom, Jui, Chien, Chuen, Tuk (female), Tuk (male), Sang+, Eed, Maew, Hook, Kuad, including Boonlerd (66), To (66), P’Kang+ (NuMap), and many many more that I just cannot list all of them here.

• I thank “Noobeed”, the Interactive Geomatic Object Oriented Language, for being software that I use throughout my research. Please visit www.noobeed.com.
• I thank Prof. Anton Schenk who is my advisor and initiate this research topic and helps get it done at the final stage. The first meeting with him was very impressive and made me switch from GIS program to photogrammetry.

• I thank Prof. Ayman Habib for his excellent advising in analytical photogrammetry. He is very friendly and always available almost anytime all year round.

• I thank Dr. Beata Csatho for serving in my dissertation committee.

• I thank Prof. William M. Masters for being my graduate faculty representative.

• I thank Ms. Irene Tesfai, Ms. Porsiri Muangsamai and the Writing Center of OSU for the correction of English grammar and usages.

• I thank the Rochenstein family, especially Pete and Kathy, who picked us up at the airport when we arrived, gave us accommodation and food, took me to school and helped us settle down in the US.

• I thank my father (Yoksia) and my mother (Kageo) who give me my life and good education. I thank my sisters and brothers (Suree, Chailai, Paiboon, Pisan and Boonsong) for taking care of my houses during my study.

• Finally, I thank my wife (Duangrat), my four daughters (Duangrudee, Rattanawadee, Boonyarat and Boonduang) and my little son (Boonoum) for being parts of my life. We came to the US and have had a wonderful time staying here for 4 years. We all love USA and this is such a beautiful place to stay. However, it is time for us to go back to Thailand and do something good for our country. So long USA.
VITA

December 9, 1960 ...................................................Born in Bangkok, Thailand

1984 ........................................................................B.Eng. Surveying,
Chulalongkorn University,
Thailand

1988 ........................................................................Postgrad Diploma (Photogrammetry)
ITC,
The Netherlands

1995 ........................................................................M.Sc. (Surveying and Mapping)
Curtin University of Technology,
Australia

1984 ........................................................................Surveyor
C.B. Surveying Co. Ltd.,
Thailand

1985-1995............................................................Surveyor and Photogrammetrist
Royal Irrigation Department,
Thailand

1996-present...........................................................Lecturer
Faculty of Engineering,
Kasetsart University
Thailand
PUBLICATIONS


FIELDS OF STUDY

Major Field: Geodetic Science and Surveying
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Background</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Motivations</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>Assumption and declaration</td>
<td>7</td>
</tr>
<tr>
<td>1.4</td>
<td>Statement of problems</td>
<td>8</td>
</tr>
<tr>
<td>1.5</td>
<td>Research objectives</td>
<td>8</td>
</tr>
<tr>
<td>1.6</td>
<td>Research methodology</td>
<td>9</td>
</tr>
<tr>
<td>1.7</td>
<td>Organization of the report</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Literature review</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>Linear feature extraction</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Feature-based matching and surface reconstruction</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Grouping 3D linear features</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>Summary</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>Study area and data set</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Study area</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>Data set</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>Image orientations</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Image normalization</td>
<td>25</td>
</tr>
</tbody>
</table>

Abstract ..................................................................................................................... ii

Dedication ................................................................................................................. iv

Acknowledgments .................................................................................................... v

List of Tables .............................................................................................................. xi

List of Figures .......................................................................................................... xii
4. Feature extraction and 3D line reconstruction ................................................... 30
4.1 Linear feature extraction ...................................................................... 30
4.2 Procedure to reconstruct linear features ..................................................33
4.3 Monotone edges.................................................................................... 34
4.4 Edge trimming .............................................................................. 35
4.5 Edge reconstruction .............................................................................. 35

5. 3D line preprocessing ....................................................................................... 40
5.1 Segment to straight lines ...................................................................... 40
5.2 Grouping ground and non-ground lines .................................................44
5.4 Line linking ........................................................................................... 45

6. Plane finding .................................................................................................... 49
6.1 Surface and planes ............................................................................... 49
6.2 Spatial searching distance .................................................................... 51
6.3 Planes from lines .................................................................................. 52
6.4 Testing for physical plane using knowledge from images ......................55
6.5 Plane fitting by least-squares adjustment .............................................. 59
6.6 An algorithm to handle UTM coordinates ..............................................60
6.7 Grouping lines to existing planes .......................................................... 62
6.8 Results of plane finding ....................................................................... 64

7. Surface representation by planes ................................................................. 66
7.1 Plane boundary ..................................................................................... 66
7.2 Recover plane boundary from line intersection ......................................68
7.3 Recover plane boundary from plane intersection ....................................68
7.4 Recover plane boundary from line in one of the images ....................... 72
7.5 Extracting more features from image ..................................................... 74
7.6 Plane formats ....................................................................................... 76
7.7 2D polygon plane format ..................................................................... 77
7.8 VRML plane format ............................................................................. 79

8. Discussion ........................................................................................................ 83
8.1 The big picture ..................................................................................... 83
8.2 Advantages of surface represented by planes ........................................ 87
8.3 Advantages of grouping features in 3D ................................................. 89
8.4 Problems .............................................................................................. 92

9. Conclusion ....................................................................................................... 94
9.1 Future work .......................................................................................... 97
9.2 Recommendations ................................................................................ 98

Bibliography ................................................................................................................100
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Values of parameters and numbers of edges found by Canny edge operator</td>
<td>31</td>
</tr>
<tr>
<td>6.1</td>
<td>Fitted plane and parameters</td>
<td>64</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Degree of complexity in digital photogrammetry tasks</td>
</tr>
<tr>
<td>3.1</td>
<td>The Ocean City study area</td>
</tr>
<tr>
<td>3.2</td>
<td>Information of the Ocean City aerial camera</td>
</tr>
<tr>
<td>3.3</td>
<td>Interior orientation measuring data</td>
</tr>
<tr>
<td>3.4</td>
<td>Exterior orientation parameters</td>
</tr>
<tr>
<td>3.5</td>
<td>Non-epipolar geometry images (a) and epipolar geometry images (b)</td>
</tr>
<tr>
<td>3.6</td>
<td>Computing normalized images</td>
</tr>
<tr>
<td>3.7</td>
<td>Original images and normalized images of the left and right photos</td>
</tr>
<tr>
<td>4.1</td>
<td>Edge images (top), matched edge images (middle) and matched edges on top of the original images (bottom)</td>
</tr>
<tr>
<td>4.2</td>
<td>linear features reconstruction flowchart</td>
</tr>
<tr>
<td>4.3</td>
<td>Monotone and Non-monotone edges</td>
</tr>
<tr>
<td>4.4</td>
<td>Edge trimming</td>
</tr>
<tr>
<td>4.5</td>
<td>Vertical and horizontal edges</td>
</tr>
<tr>
<td>4.6</td>
<td>Monotone and Non-monotone edges</td>
</tr>
<tr>
<td>4.7</td>
<td>Reconstruction of horizontal edge (top) and vertical edges (bottom)</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Photogrammetry has long been recognized as one of the most efficient techniques to make maps. Almost all national based maps are created by means of photogrammetry. Since its inception dating back to more than 100 years ago, photogrammetry has evolved by means of new technologies. Each evolution affects the way a photogrammetrist thinks and works. The three major periods consist of analog photogrammetry, analytical photogrammetry and digital photogrammetry. It is very interesting to note that the transition from the first era to the second era, and from the second to the third era, are caused by the same thing, that is, the development of computer technology. It is the computer technology, which is used to make an analytical plotter that marks the beginning of the second generation of photogrammetry. The very same technology used to build a digital photogrammetric workstation changes the concept of using analog photos to digital photos.

While the analytical plotter offers an automatic Py (y-parallax) elimination, digital photogrammetry has a very promising success in automatic Px (x-parallax)
determination. Thus terrain reconstruction can be eventually extracted from two overlapped digital images automatically. Digital photogrammetry has both images stored digitally in the system, making image analysis possible inside a computer. It is predicted that digital photogrammetry will take over analytical photogrammetry in the near future (Schenk, 1996, Schenk, 1998).

According to years of experience of the author, among all tasks in photogrammetric mapping, there is no task that is more time consuming than map compiling. This is the task where a human operator has to put a lot of effort and knowledge into his work.

Digital photogrammetry has set forth a new challenging problem, which classical photogrammetry has never thought about, that is to replace the human operator in detecting and recognizing objects. This is known as “object recognition and image understanding”. One of the golden goals addressed in Schenk (2000) is to build a map machine that could compile maps automatically.

Practically map compilation consists of collecting horizontal information and vertical information. While horizontal information includes all terrain objects and planimetric details of topography, vertical information includes terrain surfaces and mostly represented by DEM (Digital Elevation Model), or contour lines in analog maps.

In regards to manual map compilation, most photogrammetry operators, including the author, will accept that collecting vertical information in general is more tedious and more difficult than horizontal information. Fortunately, the situation is the opposite in digital photogrammetry, in which collecting terrain surface information, or surface
reconstruction, is much less complicated, therefore easier, than horizontal details.

According to Schenk (2000), shown in Figure 1.1, surface reconstruction and automatic
DEM generation, though poorly understood, are less complex than object recognition and
scene interpretation.

It seems that the first success of digital photogrammetry to replace human
operator is in the area of surface reconstruction, which happens to be the most wanted
function in terms of human operator compiling. Automatic object recognition and scene
interpretation will certainly emerge in the future.

1.1 Background

The backbone of automatic surface reconstruction is automatic matching. It
consists of two main techniques, namely, area-based and feature-based matching. In
general, linear feature matching is considered more reliable, and more meaningful, than area based matching. This is due to the fact that a linear feature in an image carries more information than a single pixel does. In some situations, for example, images of urban areas, where there exist a lot of geometric objects, extracting features for matching might be more suitable than doing area-based matching.

At the beginning, feature-based matching might be viewed as an alternative method to area-based matching. However, nowadays, features in images stand out as important objects themselves. More and more research focuses on features because they are the first meaningful information derived from images that are believed to carry enough information for other later activities of computer vision and image understanding.

Among all types of features extracted from images, the linear feature is the most interesting one. It has been the main type of feature receiving the most attention by research in photogrammetry at the Ohio State University. At least seven PhD dissertations are devoted specifically to solve problems in linear features on images: Greenfield (1987), Zilberstein (1991), Habib (1994), Cho (1995), Doorn (1991), Shahin (1994) and Zahran (1997). In addition there are more papers from OSU on linear features than those of any other types.

Recently Habib et al. (2000) used linear features in images and proposed a new matching technique, called Modified Generalized Hough Transform (MGHT), which will solve orientation and partly matching at the same time. Solving for ambiguity, by introducing pruning techniques using various types of clues, is later presented in Habib et
al. (2002) so that it is possible to have a set of 3D linear features in object space after doing MGHT.

All these examples have raised an important issue, what is going to be done next after obtaining 3D linear features in the object space?

It should be mentioned here that 3D linear features in object space do not only result from feature matching. There are other ways to obtain 3D linear features. One way is to project edges from one aerial image onto a surface in the object space. In this case, the terrain surface must be known so that light rays from the image can intersect the terrain. Nowadays it is not difficult in to get an accurate terrain surface by using laser scan technology. An algorithm to project linear features from an image, 2D space, onto a surface in 3D object space is called ray tracing. It is presented in Chapter seven of this report, in a special case when the surface is a plane.

Another way to obtain 3D linear features is from direct digitizing, by using a 3D digitizing device, which has been used widely in computer vision to capture objects in 3D space. Though it might not be relevant in photogrammetry, it does provide a conceptual vision that 3D linear features do not have to come from imagery. Hence, surveyors can use traditional field surveying equipment to collect linear features in the terrain.

It has been shown that it is possible to obtain 3D linear feature data without much difficulty. However 3D linear features are definitely not enough to represent terrain surface. They are not yet the end result in terms of digital photogrammetry goals, automatic surface reconstruction, object recognition and scene interpretation.
1.2 Motivations

Photogrammetry has learned that seeing in stereo needs more than one image. Because the human being has two eyes, there are always two images of the same objects and this is the concept of perceiving stereo. Since the past, photogrammetry has known how to utilize 3D visualization in mapping and object recognition. Aerial photography is always taken such that there are at least 60% of overlaps between photographs, and 30% of sidelaps between photo strips, to allow stereoscopic vision of all areas of coverage.

Regarding 3D data, photogrammetry has realized the importance of 3D interpretation and will use 3D visual interpretation whenever possible. An example is in photo control surveying, where a surveyor uses a pocket stereoscope to view a point on images in order to help him locate and identify the point on the ground. The author has seen many surveyors that have developed an ability to view stereo photos in the field without using a viewing tool.

In the new era of digital photogrammetry, where object recognition is one of the quests, Schenk and Toth (1991) have proposed the same above concept and in 1991 have already presented an object recognition system, called the OSU Surface Construction System. The main idea of the system is to construct a 3D surface before doing object recognition and later intelligent activity. A detailed explanation of the system can be found in Schenk (1995).

All these evidences support that object recognition, including all other activities related to object recognition and image understanding, which are to be done by a computer, must be carried out in 3D, not 2D. In other words, research should focus on
those already reconstructed 3D linear features to extract a higher level of object representation, leading to image understanding and scene interpretation. It is this motivation, to make more use of 3D linear features in the object space that drives the research that follows.

1.3 Assumption and declaration

It is true that the matching or correspondence problem in computer vision terminology, is not completely solved; feature extraction is still not perfect. In fact, at present they are still some of the most challenging problems in both computer vision and photogrammetry. This does not prevent researchers from taking a step further, which is to process and analyze 3D linear features in the object space. In fact, thinking ahead into the future is always necessary. This is what this research tries to do, based on the assumption that the results of feature extraction and 3D reconstruction of linear features are good enough to proceed to the analysis of data in the object space, regardless of whether or not feature extraction and matching are perfect.

In the computer vision point of view, this research is considered new. Much computer vision research focuses on utilizing features in 2D image simply because 2D linear features are easier to access than 3D linear features. Few studies utilize 3D data and most of them are limited to 3D points only.

In the photogrammetric point of view, this research is also considered new because most research in automatic matching using linear features stops after the matching is solved. Research in automatic 3D data organization is limited to points and it
has just started when laser scan data become more accessible. The utilization of 3D linear features will be complementary to 3D points from laser data.

1.4 Statement of problems

1. There is not enough research regarding how to handle 3D features in object space, in terms of preprocessing, grouping, and higher level surface representation.

2. There is not enough research regarding the relationship between extracted knowledge in object space and that in image space.

3. There is not enough research regarding how to compensate for and get improvements from the imperfection of feature extraction and matching algorithms.

4. There is not enough research regarding how to represent a surface derived from using 3D linear features in a better way than conventional techniques, for example, DEM (Digital Elevation Model) and TIN (Triangle Irregular Network).

1.5 Research objectives

1. to develop a basic general working scheme for constructing a surface from 3D linear features in the object space,

2. to develop a basic grouping scheme, segmentation, of 3D linear features,

3. to develop a higher level of terrain surface representation, derived from 3D linear features, that is suitable for object recognition and scene interpretation, and
4. to demonstrate the possibility of incorporating knowledge of 3D features in the object space into the image domain to help balance the imperfection of feature extracting algorithms in the image space.

1.6 Research methodology

According to the research objectives, the following methodology is proposed.

1. An ordinary widely accepted feature extracting algorithm is to be used to extract linear features from images. However, no further processing of features is to be done in the image space; rather all linear features are to be used to construct 3D linear features in the object space. A matching algorithm is considered to be perfect, or at least to the level of being good enough to construct a reasonable amount of 3D data in the object space,

2. Basic experiments grouping 3D linear features are to be carried out. They include line segmentation, line classification, line linking, and line grouping. The information drawn from the grouping process will be enough to construct planes or other higher abstract levels of data that are suitable to represent a surface.

3. Terrain surface is to be represented by a higher level abstract of data, for example, planes. Explicit information about the surface, for example, break-lines, is to be part of the surface representation.

4. Relationship between image and object space is to be exploited to help gain more knowledge on terrain surface whenever needed.
1.7 Organization of the report

This report is divided into nine chapters. The first chapter is this chapter and is considered the most important chapter where the problem is addressed and the research objectives and methodology are proposed. Chapter two will provide reviews and previous work related to this research. In short the first two chapters are intended to support the significance of the current study.

Chapter three will give background information of the study area and the data set used in the research. Chapter four is where the real work begins. It describes how this research obtains the raw data, the 3D linear features. They are to be preprocessed in Chapter five, which includes line segmentation, line classification into ground and non-ground lines, and line linking.

The results coming out from Chapter five are straight lines, which are to be used to find planes in Chapter six. Techniques used to verify whether a plane is a physical one are presented in this chapter. Then the plane boundaries are to be determined so that they can be used to represent real terrain surface. Missing boundary lines are recovered by incorporating knowledge in the image domain. Finally, two formats to describe planes that represent terrain surface are proposed. All these are in Chapter seven, the largest chapter in the report.

Chapter eight discusses the advantages and the problems of representing a surface by planes. Chapter nine restates the problems and objectives of the research. Future works are addressed followed by a concluding remark.
CHAPTER 2

LITERATURE REVIEW

This chapter reviews important topics related to this research, namely linear feature extraction, feature based matching, and 3D line grouping. Many evidences demonstrate that automatic reconstruction of linear features in 3D space is possible, although it may not be completely perfect. Thus, 3D linear feature data can be accessed without a great effort. On the other hand, grouping 3D data, in particular linear features, is a new area of research. Most research studies are done in the context of building extraction either automatic or semi-automatic. While algorithms in feature extraction and matching are more or less mature, algorithms for grouping 3D linear features are not well established.

2.1 Linear feature extraction

Linear features in the image are lines or curves showing up in the image due to rapid changes in the brightness intensity. Linear features are often called edges and they...
are normally detected by using edge operators, a process in which edges are extracted from an image. Searching for edges is one of the most fundamental and important tasks in image processing, for edge information may lead to a higher level of image processing, for example, object recognition and image interpretation.

One of the early attempts to extract edges from images is by using the Prewitt operator (Prewitt, 1970). This edge operator is a small $3 \times 3$ window mask, a so-called kernel, used to estimate the first derivative of image brightness function.

Marr and Hildreth (1980) propose the use of the Laplacian of the Gaussian (LoG) to detect edges. It is the second derivative of a Gaussian filter. The most characteristic feature of the operator is that it does two jobs in the same time, namely, smoothing the image and determining second gradient of brightness. A zero crossing searching is used to find edges, which is a set of closed contours. Due to the property of being closed contours, much research has utilized closed contours from LoG in the later task of image processing, for example, matching (Greenfield and Schenk, 1989; Hattori and Murai, 1989).

Canny (1983) proposes what is regarded as one of the most popular edge detectors, the Canny edge operator. It is in fact a sequence of four major operations, which are image smoothing by Gaussian filter, image calculation for the first derivative, non-maximum suppression in the direction perpendicular to the edge, and edge linking by hysteresis. The last step consists of two thresholds, the upper and lower thresholds. Any pixel that has its gradient larger than the upper threshold is immediately accepted as an edge pixel and is called a strong edge. Pixels with gradients less than the lower threshold
are considered non-edge. For those having gradient values in between the two thresholds, they will be considered edge, called weak edge, so long as they are connected to strong edge pixels.

In the past, a number of edge detectors have been proposed, for example, Venkatesh and Owens (1989), Venkatesh (1990), Perona and Malik (1990) and Rosenthaler et al. (1992). Their algorithms utilize the computed local energy, determined in the frequency domain. One of the newly developed edge detectors is the Susan edge operator (Smith and Brady, 1997). To get isotropic responses, it employs a circular mask, size varying from 3 to 37 pixels. The Susan edge operator can be used to find both edges and corners. It does not use image derivatives, hence is more tolerant in the presence of noise.

Another new algorithm to detect edges is from Garnica et al. (2000). They used a filter, called the maximum homogeneity neighbor filter, to smooth an image to remove noises, while preserving important image structure, for example, edges and corners.

In fact, feature extraction is a purpose driven oriented task. Recent research, for example, Dare and Dowman (2000) show that more than one algorithm of feature extractions must be used in order to improve the quality of image matching and registration.

Over the years, we have seen that there are still developments of edge detectors going on somewhere. This is mostly due to the fact that existing edge detectors are somehow not yet fully satisfactory. Moreover most edges are designed to be good in detecting one type of edge, for example, step edges. There has not yet been a universal
edge detector that can detect all types of edges. It is predicted that the quest for a good universal edge operator might not be achieved in the near future. Thus image processing must keep bearing with what is available at the time.

2.2 Feature-based matching and surface reconstruction

Surface reconstruction requires solving the matching problem, a problem that is regarded as one of the most difficult problems in computer vision society and has been the target of researchers for a long time. The most difficult part is to do the matching automatically without human interruption. In digital photogrammetry, automatic matching has been the focus since the beginning of digital photogrammetry (Schenk, 1999). Nowadays, it is generally accepted that there exist two different types of automatic matching, namely, area-based and feature-based matching.

The motivation of shifting from area-based to feature-based matching can be seen in Schenk and Toth (1991), where an excerpt is presented below.

"...Photogrammetrists are perhaps too entrenched in the micrometer and sub-pixel world, deeply if not exclusively concerned with accurate point positioning. We have to shift the focus of our attention from data-driven, pixel-to-pixel operation to a more symbolic processing of abstract representations."

Fürstner (1986) reported success in using features in automatic matching. Also Hahn and Förstner (1988) is another early success in feature-based matching. Greenfield
and Schenk (1988) have used edge-features, closed contours, found by LoG for their automatic matching. In 1990, Li and Schenk reported that they used a technique that utilizes the ψ - s representation for matching edges. And so do Schenk et al. (1991), where the multi-scale technique was used to match edges, detected by LoG. Greenfield (1991) reported a feature-based matching algorithm using an operator oriented matching system.

However Brockelbank (1991) has done an experiment of matching in SPOT images, and reported flaws in edge-based matching. He argued that, because edge finding is introduced as an intermediate step, edge-based matching is less accurate than that of area-based. Nowadays, it is the author’s opinion that the comparison between the two matching techniques is no longer an issue. In fact both compliment each other and have different purposes.

The development of feature-based matching is not yet over and still keeps going on. One example is to utilize a parallel processor in an attempt to achieve fast stereo feature matching, presented by Koschan and Rodehorst (1995). The matching problem in aerial photographs, particularly using linear features, is not only limited to the photogrammetry community. Research in computer vision often uses aerial photographs to demonstrate their matching algorithm. One example is Atalay and Yilmaz (1998), in which success of a matching algorithm is reported using linear features in aerial images.

One benefit of feature-based matching addressed by Busch (2000) is linear features can be directly used to match Geographic Information System (GIS) data, which are generally represented by features.

Newly developed matching techniques include Generalized Modified Hough Transform (Habib, Kelly, and Asmamaw, 2000). It solves the matching and orientation parameters at the same time, using accumulator arrays in the parameter space. To prevent computation explosion, an iterative procedure is utilized so that one parameter, one dimensional accumulator array, is solved one at one time. An extension of the same work is Habib, Lee and Morgan (2002). Here several pruning techniques are introduced to help solve the ambiguity problem, arising along epipolar lines.

It is worthwhile to note here that the work done by Wang (1999) is very similar to this dissertation. However, the work of Wang (1999) employed a surface image, from
LIDAR, as its main raw data, while this dissertation adopts the results of feature-based matching, more precisely 3D linear features, as the raw data.

2.3 Grouping 3D linear features

There are fewer studies on grouping of 3D linear feature than those of 2D linear feature, largely due to the following reasons:

1. Unlike point data, 3D lines data in object space is not easy to obtain.
2. There is a lack of a system to handle the 3D lines, for example, visualization, simple processing, etc. There is no general grouping rule available (Heuel, Förstner and Lang, 2000).
3. Matching lines, though becoming more popular, is not as popular as area-based matching.
4. Matching linear features is more difficult than area-based matching.
5. When the matching is solved, reconstructing 3D features from known matched features is not a trivial task.

Guy and Medioni (1997) proposed a general scheme to segment a surface, using surface patch orientation, for finding junctions and intersection curve. The work was extended by Tang and Medioni (1998) to localizing and integrating surface discontinuity found by the previous algorithm. Amir and Lindenbaum (1998) also presented a generic method for perceptual grouping of various types of features, for example, points and lines. The method consists of constructing a graph and finding the best partition of the graph into groups.
More specific research of 3D grouping is oriented toward building recognition. Examples include Baltsavias et al. (1995), Henricsson et al. (1996), Haala and Hahn (1995), Weidner (1996), Kim and Muller (1995), Jaynes et al. (1996), and Draper (1996). Most of them have somehow made use of 3D surface from DEM, either produced from softcopy image matching or obtaining directly from Laser, SAR, to find an initial position of a building on an image. Exterior orientation parameters are known, or orthorectified images are used. Jaynes, et al. (1996) and Draper (1996), however, looked for potential buildings in the image by searching for closed polygons.

It should be mentioned here that, due to the recent availability of 3D laser points, grouping 3D point becomes one of the main research area in photogrammetry, for example, Lee and Schenk (2000).

Suveg and Vosselman (2000) have shown that by integrating knowledge from image and GIS maps, the complexity of the building reconstruction process is greatly reduced. For a semi-automatic approach, Rottensteiner (2000) presented a system that let an operator identify an initial position of the building in the object space. Then the system would precisely find the building by utilizing stereo-photos along with rigorous photogrammetry mathematical model.

Heuel, Förstner and Lang (2000) state that grouping straight lines and planes in 3D often appear in the context of building extraction, and that no general grouping rule has been established.
2.4 Summary

It is seen that feature extraction from an image has a long history, dating back to 1970. This area of research can be considered mature. However, current evidences of image processing research inform that feature extraction, in particular linear features, need improvement and is still being improved.

Matching linear features is a difficult problem in computer vision and it has been attacked over the past years. Many recent studies, from both photogrammetry and computer vision, have demonstrated the success of using linear features for solving the matching problem.

Grouping 3D data, in particular linear features, is a new area of research. Most research is done in the context of building extraction, either automatic or semi-automatic. While algorithms in feature extraction and matching are well established, algorithms for grouping 3D linear features are not well established.

Moreover, research on building extraction has somehow made use of the terrain surface, for example, DEM or DSM (Digital Surface Model), in order to extract buildings from 3D data. This basically contradicts the purpose of this research where a surface is in fact unknown and is the final target information that is to be derived from 3D linear feature data.
CHAPTER 3

STUDY AREA AND DATA SET

This chapter provides a general view of the study area and the working data set, which are basically aerial photographs. Results of image orientation are reported. It is worthwhile to note that, although there exist many systems that can automatically extract and reconstruct features in aerial images, such systems are mostly in research institutes and not yet commercially available at the time of this research. In order to obtain a data set of 3D linear features in object space, the author has made use of a number of tools available at the time at OSU, including a technique called image normalization. Thanks to the algorithm of image normalization generation given in Schenk (1999), the effort of reconstruction of linear features is greatly reduced.

3.1 Study area

The study area is in Ocean City, Maryland, U.S.A., located approximately at latitude 38° 20' N and longitude 75° 05' W. It is a completely urbanized area, about 80 %
Figure 3.1 The Ocean City study area. Shown at the top is an aerial photo of the study area over Ocean City. The bottom picture is the enlargement of the working area.
of its area are identified as buildings, and others are roads and parking spaces. The terrain is rather flat with an average elevation of -35 meters (WGS84 height).

The working area coverage is 100 x 130 meters, shown in Figure 3.1. Most of the area is covered with buildings (apartments, houses, office buildings) and some trees. Cars are seen parking in parking lots throughout the area.

The study site is about two blocks of streets in Ocean City, and it contains more than 20 buildings. Due to the large scale of the photographs, buildings are easy to identify. By visualization in an analytical plotter, all roof- tops of buildings in the working area are planar surfaces. Some of which are of simple flat roof type and some are complicated, consisting of more than 3 planar surfaces.

3.2 Data set

The data set consists of a set of aerial photographs taken by an aerial camera, whose information is described in Figure 3.2. The aerial camera is of wide angle type with a focal length of 152.74 millimeters. The mean terrain elevation, WGS84 ellipsoid height, is -35 meters. and the aircraft altitude is 540 meters. Thus, the approximate photo scale is 1:3750.

Analogue photographs are converted to the digital form by using a high resolution geometric scanner at the Byrd Polar Research Center of the Ohio State University. Scanned at a resolution of 15 micrometer, gray scale, one digital image acquires an
The reference side is the side with the camera name and photo number on it. The ref side is oriented so that it is on the left hand side. ID starts from 99991 at upper left and up left to right and top to bottom.

<table>
<thead>
<tr>
<th>ID</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>99991</td>
<td>-106.001</td>
<td>106.008</td>
<td>0.003</td>
</tr>
<tr>
<td>99992</td>
<td>-105.993</td>
<td>106.000</td>
<td>0.003</td>
</tr>
<tr>
<td>99993</td>
<td>-105.993</td>
<td>-0.001</td>
<td>0.003</td>
</tr>
<tr>
<td>99994</td>
<td>-105.993</td>
<td>-0.001</td>
<td>0.003</td>
</tr>
<tr>
<td>99995</td>
<td>-105.993</td>
<td>106.006</td>
<td>0.003</td>
</tr>
<tr>
<td>99996</td>
<td>-105.993</td>
<td>-0.001</td>
<td>0.003</td>
</tr>
<tr>
<td>99997</td>
<td>-105.993</td>
<td>-0.001</td>
<td>0.003</td>
</tr>
<tr>
<td>99998</td>
<td>106.016</td>
<td>105.997</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Figure 3.2 Information of the Ocean City aerial camera

The small patch of study area extracted from the original image has a size of about 3 x 3 centimeters on the photo. The image size of the study area is about 3 megabytes, 1600 rows by 1900 columns.

It should be mentioned here that the complete data set, in fact, consists of several types of sensor data, namely airborne laser scanner data, hyper-spectral data, aerial photographs, satellite imagery (SPOT and Landsat) and GPS. It is a project conducted at OSU to study an integration of many different sensor data (Csatho and Schenk, 1998).

approximate size of 16K x 16K pixels, about 300 megabytes. The two aerial photos that cover the study area are photo numbers 6176 (left) and 6175 (right).
3.3 Image orientations

Interior orientation (IO) of the two aerial images, 6176 and 6175, was performed digitally by measuring row and column numbers of fiducial marks. Shown in Figure 3.3 is the raw measurement data of IO. The accuracy of the measurement is set to 0.3 pixels, while that of fiducial marks is 3 micrometers. By using the affine transformation, a standard deviation of unit weight of about 1.5 is achieved in both images.

![Table of Interior Orientation Measuring Data]

Figure 3.3 Interior orientation measuring data

The Exterior Orientation (EO) of the two working photographs is done by using ground control points identified from airborne laser data. The coordinate system is UTM zone 18, WGS84 datum. There are 4 ground control points, with a standard deviation of

24
15 centimeters, used in the orientation. The computations of EO yield a standard deviation of unit weight of about 1.0 for both photos, given standard deviations of 0.015 millimeters and 0.25 meters for photo coordinates and ground control coordinates respectively. Figure 3.4 gives the values of Exterior Orientation Parameters (EOP) of photos 6176 and 6175.

<table>
<thead>
<tr>
<th></th>
<th>6176 (Left)</th>
<th>6175 (Right)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\omega)</td>
<td>-0.933946074893944 deg</td>
<td>-0.917166279790520 deg</td>
</tr>
<tr>
<td>(\varphi)</td>
<td>0.198367653734182 deg</td>
<td>0.282146115668055 deg</td>
</tr>
<tr>
<td>(\kappa)</td>
<td>73.891818031368075 deg</td>
<td>74.127544774737785 deg</td>
</tr>
<tr>
<td>(X_0)</td>
<td>492380.882 m</td>
<td>492464.976 m</td>
</tr>
<tr>
<td>(Y_0)</td>
<td>4242026.388 m</td>
<td>4242326.686 m</td>
</tr>
<tr>
<td>(Z_0)</td>
<td>544.416 m</td>
<td>540.713 m</td>
</tr>
</tbody>
</table>

Figure 3.4 Exterior orientation parameters

3.4 Image normalization

In an oriented stereopair, conjugate features will appear along epipolar lines. It is unlikely that epipolar lines are identical with the rows of images. Hence, searching for a corresponding feature may take across several rows, resulting in more computational time. In order to reduce the effort of searching corresponding objects in a stereopair image, both images must be registered in the epipolar geometry. Figure 3.5 and 3.6 describe some of the fundamental concepts in computing normalized images. Figure 3.5 shows the difference between non-epipolar geometry stereo-pair and epipolar geometry stereo-pair. Figure 3.6 describes the concept of how epipolar geometry images are
created, when EOPs are known. The main task is to rotate the two images so that their x-axes, in photo coordinate system, are parallel to the airbase. To do so, first, both images have to be rotated back to the vertical attitude, and then they will be rotated by 3 angles, namely \( \omega, \varphi, \kappa \) (see Figure 3.6).

Figure 3.5 Non-epipolar geometry images (a) and epipolar geometry images (b) (after Schenk, 1999)
The angle \( \omega \), not shown in the figure, can be selected so that the two normalized images can cover the largest area possible on ground. Epipolar geometry images are called normalized images (Schenk, 1999). More details and algorithm to compute normalized images can be viewed in Schenk and Toth (1991), Cho, et al. (1992) and in chapter 12 of Schenk (1999).

The normalized images of photo 6176 and 6175 are generated with a compatible resolution with the original image resolution (15 micrometers). Finally, the histogram matching technique was performed on one of the normalized image, 6176. This is to
Figure 3.7 Original images and normalized images of the left and right photos
make the overall appearance, in terms of brightness, of the two images as similar as possible. The original images and the normalized images are shown in Figure 3.7, in which the left normalized image with histogram matching appears at the bottom left and an example of a full resolution image patch is shown at the bottom right of the figure.
CHAPTER 4

FEATURE EXTRACTION AND 3D LINE RECONSTRUCTION

This chapter describes in detail procedures to extract and reconstruct linear features in the object space. The raw data are normalized images created by the method described in the previous chapter. The procedure to extract linear features in images is basically edge detection. The Canny edge operator was used in this research. To guarantee that the 3D features in object space are correctly reconstructed, matching is done under a human visual supervision. However, when the matching is identified, reconstruction is done automatically. Like matching, the process to reconstruct linear features in object space is also not trivial.

4.1 Linear feature extraction

Linear features can be extracted from images via an edge operator. A homegrown Canny edge operator developed at OSU is used to extract linear features in the images of the study area. There are 2 important parameters set in the Canny edge operator. One is
the standard deviation of the Gaussian filter, and the others are the two threshold values for edge binarization. Canny (1983) used two threshold values to reduce the presence of broken edges. Table 4.1 shows the values of parameters set to detect edges in photos 6176 and 6175. The numbers of edges found and matched in each image are also reported. The number of matched edges is 403, or about 12% of that of the original images. The results of edge detection, as shown in Figure 4.1, are two binary files, edge images, of the left and right photo. Also shown at the bottom of the figure are images of matched edges, which are plotted on top of the original images, shown at the bottom of the figure.
Figure 4.1 Edge images (top), matched edge images (middle) and matched edges on top of the original images (bottom)
4.2 Procedure to reconstruct linear features

The whole procedure to reconstruct 3D linear features is summarized by a flowchart, shown in Figure 4.2. Starting off by monotone edges (will define in the next section) all corresponding edges are trimmed to the overlapped region. Then they are grouped into horizontal or vertical edges. A pair of corresponding horizontal edges gets reconstructed by using the two end points, determined by an interpolation from a line equation. A pair of vertical edges gets reconstructed by computing intersection.

![Figure 4.2 linear features reconstruction flowchart](image)

Figure 4.2 linear features reconstruction flowchart

33
points for every pair of pixels in the same row. Pixels that have the same row number as an existing pixel will be removed. The following sections describe all the details.

4.3 Monotone edges

Apart from the edge images, the Canny edge operator creates a file consisting of a list of connected pixels, in rows and columns, of each individual linking edge. There are two files for each image. This information is very useful for they are raw-matched pixels that will be used to construct a set of linear features in object space. Thus, two pixels in the images, one from the left and one from the right, will generate one pixel in 3D object space. Although the matching is identified, the arrangement of pixel within the two matched edges can be different. To help increase the efficiency of reconstruction, connected pixels within matched edges are reordered so that the first pixel is the topmost pixel, having the smallest row number. Such edges are called monotone edges, and they are depicted in Figure 4.3.

![Monotone and Non-monotone edges](image)
4.4 Edge trimming

A matched edge from the left image might not have the same number of pixels as that of the corresponding edge on the right image. For those pixels that are not within the overlapping region of the two matched edges, it is not possible to use them for reconstruction; hence, they must be removed beforehand. Figure 4.4 demonstrates a situation in which the left matched edge and the right matched edge do not have equal length. Some pixels which are outside the overlapping region must be trimmed out.

4.5 Edge reconstruction

Before a pair of corresponding edges is reconstructed in the object space, its orientation with respect to the image is determined, and later will be classified as vertical or horizontal edges. This is due to the fact that reconstruction of vertical edges and...
absolute (slope) > 0.4

Monotone vertical edge

absolute (slope) ≤ 0.4

Monotone horizontal edge

Figure 4.5 Vertical and horizontal edges

point get constructed in object space
pixel

fitted straight line

Left matched horizontal edge  Right matched horizontal edge

Left matched vertical edge  Right matched vertical edge

Figure 4.6 Reconstruction of horizontal edge (top) and vertical edges (bottom)
horizontal edge will have different algorithms. While reconstruction of vertical edges can be done with much precision, reconstruction of horizontal edges needs a greater care and may be less accurate. It should be noted that reconstruction of two perfect horizontal edges is undetermined due to an ambiguity along an epipolar line, unless corresponding point features are used.

An estimation of the slope of each individual edge is computed in order to discriminate vertical edges from horizontal edges. This is done by fitting a straight line to an edge. An edge with a slope of 0.4 or greater is classified as a vertical edge; otherwise, it is a horizontal edge (Figure 4.5).

To reconstruct vertical edges, redundant pixels that share the same row as other pixels will be removed. In other words, at the particular row number, there will be only one pixel on the left image and one corresponding pixel on the right image, that get reconstructed. However, to reconstruct horizontal edges, a straight line is fitted to the two corresponding edges, then the x coordinates, or column numbers, of the two end points of the edge are computed by interpolation from the straight line equation (Figure 4.6) using two equal y coordinates (row numbers). In this way, horizontal lines will be more accurately reconstructed than just taking the two endpoint pixels.

To compute a point in 3D object space from two points in a stereopair image, the procedure is called ray intersection. A light ray from a point on the left image, passing through the left project center, makes an intersection with a light ray from the corresponding point on the right image that passes through the right projection center.
The orientation of the two images guarantees that all corresponding light rays must intersect (within the accuracy of the orientation parameters).

It should be noted that, when reconstructing, features are basically dissembled to points and are reconstructed by using ray intersecting from points, Habib et al. (2002). Finally, a set of 3D linear features in object space is obtained. Figure 4.7 shows a part of a file used to store 3D reconstructed linear features.

It is seen that linear features that are constructed from horizontal edge will have only two points, while those from vertical edges can have many points. In summary,

Figure 4.7 3D linear feature data
there are 403 matched edges reconstructed in the object space. The resulting 3D linear features will serve as a fundamental data set for surface reconstruction analysis in the next chapters.
CHAPTER 5

3D LINE PREPROCESSING

Reconstruction of linear features in images yields a set of 3D linear features, or 3D lines, in the object space. Since originally linear features in images are represented by a string of pixels, the resulting 3D lines in the object space will consist of a string of 3D points. This generates redundant points in the data set. To eliminate redundant points is one of the tasks in data preprocessing, which also includes some basic grouping, such as classification of ground and non-ground lines, grouping of horizontal lines, and line linking. This chapter describes those basic tasks related to 3D linear feature preprocessing.

5.1 Segment to straight lines

The algorithm to segment linear features to straight lines is modified from the 2D line generalization of the Douglas–Peucker algorithm. The algorithm is demonstrated in Figure 5.1. The concept is to connect two end points with a straight line, then find distances from all other points to the line. If the distance from the farthest point to the
Step 1 find the point that has the maximum offset distance

Step 2 connect points, if $D_{\text{max}} >$ tolerance

Figure 5.1 Douglas–Peucker algorithm

Figure 5.2 Distance from a point to a line in 3D
line is greater than a specific tolerance ($D_{max}$), the line is divided into two straight lines, using the farthest point as one of the end points, and the algorithm keeps repeating until there is no point with its distance greater than the tolerance. Douglas–Peucker algorithm is used widely in map generalization.

Douglas–Peucker algorithm is adopted in this research to segment 3D linear features into 3D straight lines. An algorithm to compute the distance from a point to a straight line in 3D space is taken from Rodriguez (2001). Shown in Figure 5.2 is a line in 3D, with position vector $a$ and direction vector $v$, a vector $p$, is the position vector of an arbitrary point in 3D. The angle between the vector $v$ and vector $p-a$ can be calculated from the dot product of the two vectors. Finally the formula to compute the distance from a point to the line is expressed by $|p-a| \sin \theta$.

By using Douglas–Peucker algorithm, the original 403 lines are segmented into 481 straight lines in the object space. Figure 5.3 shows all the lines plotted on top of the original image 6175, with point IDs next to them. Some of the ID numbers might be illegible due to a big reduction of scale. A special system of line ID is introduced here so that, by looking at the line ID, it is possible to tell whether a line is from another line. This is done by adding 3 more digits to the previous line ID. The reason to use 3 digits is to avoid ambiguity, because the number of lines in the original set is 403. Examples of new line ID are 30001, 30002 and 30003, which are segment from the original line ID 30.
Figure 5.3 Segmented lines and ID

Figure 5.4 Histogram used to group ground and non-ground lines
5.2 Grouping ground and non-ground lines

The procedure to grouping lines into ground lines and non-ground lines is by using histogram binarization. To construct a histogram, each single line will give a vote in a category according to its elevation. The bin size is set at 1 meter interval. By visualizing the histogram (shown in Figure 5.4), it is found that the histogram consists of mainly two groups of object, ground lines and non-ground lines.

As shown in Figure 5.4, the two groups can be clearly divided at an elevation of -33 meters. Thus, the elevation of -33 meters is used to segment lines into 2 groups, ground lines and non-ground lines.

Of the 481 line data set, there are 119 lines that are classified as ground lines, and 362 lines that are non-ground lines. They are back projected on to the original image, photo 6175 (Figure 5.5).

Figure 5.5 Ground line (left) and non-ground lines (right) on the original image
It is found that some ground lines that look like parts of a building, for example, lines around the top right building of the left picture in Figure 5.4, are in fact shadows of the building. This area is enlarged and plotted in full resolution in Figure 5.6. It is clearly seen that they are not parts of the building, and they can be only detected by using 3D data.

5.3 Line linking

The purpose of line linking is to eliminate broken lines that might occur due to imperfect feature extraction in image space. Broken lines in object space must be linked
if they represent the same physical line. To eliminate them, 3D lines in the database are searched exhaustively one by one to find out whether there are lines that lie exactly on it. Figure 5.7 shows the concept of line linking.

Theoretically, this is an \( N \times N \) complexity operation. However, since a set of broken lines will be most likely close to each other, a criterion can be set so that searching is done only within a certain vicinity of broken lines. Hence the complexity can be reduced significantly, especially when there are a large number of lines in the database.
To combine broken lines in the object space, knowledge in image space is utilized. One piece of information can be from the original edge images. It is possible to

Figure 5.8 Broken lines around building
project the reconstructed broken lines back onto the original edge images. Then evidences can be searched whether the broken lines are from one of a longer line in one of the images. In some cases, for example, that in Figure 5.8, it is found that the broken lines are from one of the image whose signals in the particular area are so weak that edges are corrupted by noise. By investigating the left and right edge images, lines ID 179, 180, 181, 182 and 183 are combined. And so are line ID 175, 176, 177 and 178, as well as, 184, 185 and 186.

To avoid mathematic operations in line linking, a special algorithm is designed so that there is no distance computation among points. Rather lines are linked by coordinate comparison. Finally, of the 362 non-ground lines, there are 68 lines being combined into 23 lines, shown in Figure 5.9. Thus, the total number of non-ground lines left in the database is 317 lines.

![Figure 5.9 Line linking of 23 sets of broken lines shown by circles (left) and result of line linking (right)](image-url)
CHAPTER 6

PLANE FINDING

This chapter describes two techniques that are used to find planes, which will be later on used as fundamental primitives for surface representation. The input data are segmented lines resulting from data preprocessing in previous chapters. First, a general concept to group parallel lines, intersection lines and horizontal lines are introduced. They are performed under a concept of searching distance and defining a spatial volume to search for plane at a particular line. Knowledge in image space is utilized so that the planes found are representations of real physical planes in the object space. At the end of the chapter, a special algorithm to fit a plane for points, whose UTM coordinates are large, is presented.

6.1 Surface and planes

In an abstract point of view, a point is a dimensionless object. A line is a one-dimensional object; hence, it does not have width. A straight line in 3D space can tell
nothing except the two directions to which the two ends point. A group of straight lines might be able to tell information about a surface, but explicit information must still be
drawn from them. Thus, to represent a surface with lines, a higher abstraction level is needed to extract from lines.

So far, the current data set consists of 3D straight lines in object space, classified as ground lines and non-ground lines. Those of non-ground lines are presented again, in the object space coordinate system, as shown in Figure 6.1. The upper figure shows lines plotted in 3D view. The lower one, which looks less confusing, is the projection of lines in the XY plane.

Lines shown in Figure 6.1 are non-ground lines; hence, they are man-made objects. Here, in an urban area where there are a lot of manmade objects, mostly buildings, it is found that the most common type of surfaces is planar surfaces, or planes. Therefore, in order to represent a surface from 3D lines, a set of planes must be extracted from them.

6.2 Spatial searching distance

Now a concept of spatial searching distance is introduced. Starting from a particular line, a plane is to be found only within a certain neighboring distance. It is called searching distance. Hence a straight line in 3D will have a searching volume that looks like a cylinder with two half-sphere caps at both ends (see Figure 6.2).

To set an appropriate value of searching distance, knowledge of object sizes is essential. By investigating the sizes of buildings in the study area, it is found that the
Figure 6.2 Searching volume at a line

largest building is no longer than 30 meters. Therefore, a searching distance of 30 meters is set for the plane searching.

6.3 Planes from lines

Fundamentally a plane in the 3D space can be described by using 2 lines that are either parallel to or intersects with each other. It is seen that not all arbitrary pairs of lines can construct planes, for they will not always parallel to or intersect with each other.
Figure 6.3 depicts three situations where 2 lines can construct a plane. They are a pair of intersection lines, parallel lines and horizontal lines. The third situation is in fact a special case of parallel lines; however, it may be the most popular situation in an urban area.

A pair of parallel lines can be found by using the vector approach with a little computational effort. First the two lines being considered are reduced to vectors, by coordinate differencing. Then they are normalized to have a unit length. After that, the difference in coordinates is to be compared to check whether they are parallel.

To test whether a line intersects with another line can be done by projecting the two lines on to a plane, for example the XY plane, and then checking whether the two projected lines intersect. If not, the two lines do not intersect in the 3D space. If yes, the intersection point is to be projected back on the original lines to check whether they are the same point in the 3D space. The concept is depicted in Figure 6.4.

![Figure 6.3 Plane from intersection lines (a), parallel lines (b) and horizontal lines (c)](image)
Both testing for parallel lines and intersecting lines require a tolerance to be set so that a comparison can be made. Due to the error in image coordinates, reconstructed points in object space will not be perfect, but carry error propagated from image coordinates. The orientation parameters are also afflicted by errors and they can affect the quality of points in object space as well. Therefore a tolerance must be introduced when testing for parallel lines and intersecting lines.

Figure 6.4 Test of line intersecting
The rule of thumb is to multiply the error in image space with the photo scale. Since the image scale is 1:3750 and is scanned at 15 micrometer, the approximate accuracy of reconstructed points is $0.015 \times 3750 \approx 60$ millimeters, assuming that the accuracy of image coordinates is $\pm 1$ pixel. Next, the tolerance is set to 3 times of the estimated accuracy, which is considered to be the maximum likely error according to statistics. Thus, the value of tolerance used to test points in object space is set at $3 \times 60 = 180$ millimeters, or 18 centimeters.

6.4 Testing for physical plane using knowledge from images

When a pair of lines that are either parallel or intersecting is found, it may or may not be parts of a plane representing the surface of the terrain. It is wrong to construct a plane that does not exist in the object space. In other words, to check whether a plane is a physical plane is necessary. To do so, knowledge in the image domain can be utilized by extracting more information suitable to analyze plane existence. One piece of information is the homogeneity of brightness values of a plane. It is expected that pixels that belong to a plane will have similar brightness values. This is based on the fact that the incident angle, an angle between surface normal vector and light source vector, is constant all over the plane. Hence, all pixels in the plane get the same amount of energy and should reflect equal energy to the camera, provided the plane is made by same material.

In order to demonstrate the concept, two situations are presented here. The first one is shown in Figure 6.5. It shows three lines under consideration during plane finding,
Figure 6.5 Testing for physical plane from parallel lines
namely line ID numbers 58, 156 and 234. All of them are back projected onto the original image, photo number 6175. The line number 58 is the line being investigated. Two lines are found parallel to it, one on the left side (156) and the other on the right (234).

It should be mentioned here that a line in 3D space can have either one plane or two planes (the most) connected to it, depending upon whether it is a break line. In case that a line is a break line, it can have two planes connected to it. One is to the left and the other is to the right of the line.

First a polygon is created by line numbers 58 and 156. The dotted lines in the image are imaginary lines that define the boundary of the polygon. All pixels within the polygon are extracted and a histogram is computed, using a bin size of 10. It is found that the histogram, shown at the left, has one peak, meaning that all pixels are most likely from the same population, in this case, the same physical plane. At this point it can be concluded that there is a chance that line numbers 58 and 156 create a plane in object space.

On the other hand, the polygon constructed by line numbers 58 and 234, does not have a homogeneous brightness. According to the histogram shown at the right of the figure, pixels in the polygon belong to, at least, two populations, seen as two peaks in the histogram. It is concluded that lines 58 and 234 do not create a physical plane in the object space.

Another example is given in Figure 6.6. This is an example to show that both planes on either side of the line being considered are likely to be physical planes.
Line number 476 in the Figure 6.6 is an example. It has two lines intersected with it, line numbers 475 and 53. Two polygons are drawn to create a plane territory, from which pixels are extracted. The two

Figure 6.6 Testing for physical plane from intersecting lines
polygons have their histogram showing that both areas are homogeneous. Thus, it is concluded that there are two planes on either side of line number 476, one made from line number 475 and the other line number 53.

6.5 Plane fitting by least-squares adjustment

A plane equation is defined as:

\[ z = Ax + By + C, \]

where \( x, y, z \) are coordinates of a point in a plane, and \( A, B, \) and \( C \) are plane coefficients. It is seen that three points are required to solve for a plane equation.

When two lines in 3D space perfectly lie on a plane, there is one point that can be thought of a redundant. This is because three points are actually enough to construct a plane, provided they are not on the same straight line.

In reality, there is no perfect parallel or intersecting lines; thus, no perfect plane exists, in the sense that all 4 points lie exactly in one plane. In order to get appropriate values of plane coefficients, least-squares adjustment is used. The observations here are heights, \( z \) coordinates, expressed as a function of parameters \( A, B, \) and \( C \). For the sake of simplicity, \( x \) and \( y \) coordinates are treated as constants, leading to the formation of the so-called observation equations.

When fitting a plane to a pair of lines, it is basically like fitting a plane to a set of points, in this case the four end points of the two lines. However, the four points will not have an equal weight due to a difference in line lengths. A line can have many points in it. The longer the line is, the more points it has. Hence it is fair to describe weight of a
line as a function of its length. The simplest one is to set the weight equal to the line length. Figure 6.7 demonstrates the concept.

![Figure 6.7. Weight strategy in fitting a plane by least-squares adjustment](image)

6.6 An algorithm to handle UTM coordinates

UTM coordinates are known to have large values of coordinates, for its design is oriented toward a universal mapping purpose. An area that is at high latitudes will normally have big y coordinates, which are distances from the equator. The working area
in Ocean City has coordinates in UTM map projection, with an approximate value of 492,000 E and 4,242,000 N.

Having large values of coordinates creates problems in the computation of plane fitting. On investigating the plane equation, it is found that the design matrix is badly imbalanced. Each row of the matrix has three columns. The first two columns are x and y coordinates and the third one is a constant 1, as follows:

\[
\begin{bmatrix}
  x_1 & y_1 & 1 \\
  x_2 & y_2 & 1 \\
  \vdots & \vdots & \vdots \\
  x_n & y_n & 1 \\
\end{bmatrix}
\]

It is seen that the design matrix has its elements in the first two columns larger value numbers, while those of the third column are ones. The imbalance can significantly deteriorate the accuracy of computation. One way to solve the problem is to shift all the x and y coordinates by a certain constant, for example, by an amount equal to the minimum of x coordinates in the x direction. However, to do this, one has to give up the UTM coordinate system and works with a kind of local system. In other words, after adjustment, the determined plane coefficient will not valid for the original UTM coordinates. This makes another problem since one has to carry not only plane coefficients but also the values of the shift to the origin of the UTM coordinate system.
A better way to solve the problem is to divide the x and y UTM coordinates in the design matrix by their mean before an adjustment computation. This time, the results will be much more precise because the design matrix is well-balanced, having most of its element values close to one. After adjustment, the values of A and B are to be divided by their means again to recover the correct value of parameters A and B.

6.7 Grouping lines to existing planes

It is possible to group lines that belong to one plane. Once a potential plane from a pair of lines is found, it can be used to calculate whether there are more lines that lie on the plane. This can be simply done by using an algorithm to calculate the distance from a point to a plane. If the two end points of a line have the distances to the plane less than a tolerance, set at 18 centimeters, the line is on the plane.

The formula to calculate the distance from a point to a plane in 3D space is taken from Protter and Morrey (1973), as follows:

\[
d = \frac{|Ax_1 + By_1 + Cz_1 + D|}{\sqrt{A^2 + B^2 + C^2}}
\]

In the formula, \(d\) is the distance from a point \((x_1, y_1, z_1)\) to a plane, having a general equation of \(Ax + By + Cz + D = 0\). It is important to pay attention to the value of the denominator. If it is less than or equal to zero, the distance is zero. Essentially when more lines are included in the existing plane, its plane parameters can alter to reflect the majority of the observations. Hence a new adjustment is needed to fit a plane to the set of existing lines.
Another thing that gets changed when combining more lines into an existing plane is the effective boundary of the plane. As demonstrated in earlier figures, Figure 6.5 and 6.6, when testing for homogeneity of an area of a plane, a polygon is drawn around existing lines so that the boundary of the polygon is the outermost boundary of the lines. This is, in fact, the same concept of the so-called convex hull.

When adding more lines to an existing plane, the area of convex hull also changes. Figure 6.8 shows an example of grouping more lines to an existing plane. In

![Diagram of grouping lines with an existing plane](image)

**Figure 6.8.** Grouping lines with an existing plane
the figure, all lines whose IDs are surrounded by circles are lines that have been grouped into the plane. The dotted lines show the current plane boundary when projected on the original image.

6.8 Results of plane finding

There are 27 planes found in the study area, as shown in Table 6.1. The last

<table>
<thead>
<tr>
<th>Plane ID</th>
<th>No of lines</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Average residual (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>0.654615907</td>
<td>-0.296932459</td>
<td>937337.922</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>-0.561213029</td>
<td>0.230786199</td>
<td>-702772.703</td>
<td>0.06</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-0.307493454</td>
<td>-0.702854288</td>
<td>3133054.000</td>
<td>0.29</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.116029897</td>
<td>-0.251078704</td>
<td>-1122296.750</td>
<td>0.02</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>-0.399458745</td>
<td>-0.881577264</td>
<td>3936510.563</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0.302520639</td>
<td>0.870981249</td>
<td>-3843888.250</td>
<td>0.46</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>-0.951262716</td>
<td>0.450975525</td>
<td>-1448331.125</td>
<td>0.18</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>0.101401997</td>
<td>0.031394688</td>
<td>-82288.901</td>
<td>0.31</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>0.554284379</td>
<td>-0.186359455</td>
<td>517648.563</td>
<td>0.14</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>-0.576365806</td>
<td>0.366204375</td>
<td>-1269775.406</td>
<td>0.12</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>0.3285465208</td>
<td>-0.1649137884</td>
<td>538714.098</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>-0.335026502</td>
<td>0.114136596</td>
<td>-319269.055</td>
<td>0.01</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>-0.026499695</td>
<td>0.069323371</td>
<td>-281065.766</td>
<td>0.34</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>0.086797017</td>
<td>-0.030769886</td>
<td>87761.660</td>
<td>0.20</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>-0.006937095</td>
<td>0.336553897</td>
<td>-1424351.875</td>
<td>0.28</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>0.015820691</td>
<td>0.075017061</td>
<td>-326058.813</td>
<td>0.24</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>0.159411330</td>
<td>0.615968220</td>
<td>-2691595.453</td>
<td>0.48</td>
</tr>
<tr>
<td>17</td>
<td>6</td>
<td>0.287865545</td>
<td>-0.075899164</td>
<td>180206.953</td>
<td>0.29</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>0.795743591</td>
<td>-0.354775273</td>
<td>111314.438</td>
<td>0.54</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>0.151248884</td>
<td>0.462044853</td>
<td>-2024567.750</td>
<td>0.17</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>0.000758640</td>
<td>-0.029608211</td>
<td>125201.570</td>
<td>0.01</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>0.052063385</td>
<td>-0.19617506</td>
<td>806545.902</td>
<td>0.34</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>0.421314487</td>
<td>-0.012287358</td>
<td>-155331.344</td>
<td>0.08</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>0.4096500967</td>
<td>-0.1801832190</td>
<td>562633.625</td>
<td>0.02</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>-0.5155591637</td>
<td>0.185377966</td>
<td>-532402.812</td>
<td>0.05</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>0.2309796732</td>
<td>0.3519762359</td>
<td>-1606897.188</td>
<td>0.12</td>
</tr>
<tr>
<td>26</td>
<td>119</td>
<td>0.0046483097</td>
<td>-0.0153382174</td>
<td>62743.610</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Table 6.1 Fitted plane and parameters

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
plane, ID 26, is the ground plane. All other are non-ground planes. Figure 6.9 shows all non-ground planes with their ID plotted on the original image. These planes and parameters will be used for further processing in the next chapter.
CHAPTER 7

SURFACE REPRESENTATION BY PLANES

This chapter focuses on the final step regarding surface reconstruction using planes, which are derived from linear features in the object space. First, the essence of finding the plane boundary is addressed. It is followed by three methods of recovering plane boundaries. The first two use plane and line intersection without knowledge of the image space. The last one is assumed to have a partial knowledge of one of the images of the stereopair. An idea for extracting more features from the image domain is presented, using existing planes and boundaries that were already found. At the end of the chapter are two proposed efficient plane formats that could be used to represent a surface.

7.1 Plane boundary

In mathematics, planes do not have boundaries. They extend infinitely everywhere throughout the space. However, physical planes in reality are finite, and do have boundaries. In order that planes can be used to represent a surface, their boundary must be known. This is particularly useful when one wants to find out on which
particular plane a point on a surface is. A complicated surface may consist of several planes, and plane boundaries are obviously lines where two planes intersect. In this case, they are actually break-lines.

From last chapter planes are described by using the plane equation. They do not have information about the boundary. Now, those planes are to be investigated again so that the plane boundaries are known. In order to do this, it is necessary to set down some assumptions as follows.

- The plane data in this research are special for they are generated from straight lines, which are derived from changes of brightness in the images. It is reasonable to infer that straight lines from which a plane is formed are potential plane boundary, or break lines.

- The plane data in this research are generated in such a way that lines belonging to the same plane are already grouped together. It is reasonable to infer that the outermost boundary of straight lines, from which a plane is formed, is the plane boundary.

- The plane data in this research are non-ground planes for they are generated from non-ground lines. In addition, they are checked for being real physical planes, hence associated with physical objects in the object space. Since the scene is a vertical aerial scene of urban area, it is reasonable to infer that most planes are most likely from buildings, which are manmade objects. Thus the plane boundary is limited to geometric polygonal shapes, such as triangle, rectangle, etc.
• In order to connect manmade objects to the ground, non-ground planes that are not connected to the ground surface will be connected to the ground surface via vertical planes.

It is seen that all of the above assumptions are actually based on the natural properties of the data, which result from previous data processing. Apart from the utilization of a little knowledge of the scene for the last two assumptions, there is not any newly introduced restriction in terms of availability of data and domain knowledge.

7.2 Recover plane boundary from line intersection

Boundaries of planes that are generated from intersecting lines are to be extended so that intersection points become new corners of the plane boundary. The concept is shown in Figure 7.1, in which a plane is generated by line 1, line 2 and line 3. The current boundary is the outermost of the lines, shown as the shade area in the figure. Two new corners of the plane boundary are computed by intersection of lines 1 and 2 and lines 2 and 3. Consequently, a new plane boundary is created, shown as dotted lines in the figure.

7.3 Recover plane boundary from plane intersection

When two planes intersect in space, a break-line is generated. Due to different orientations of planes in the object space, most break-lines will show up as edges in the...
images because of changes of brightness from plane to plane. However this is not always the case. Sometimes break-lines do not show up on both images because of low image contrast. In this case, the edge operator will fail to detect break-lines in both images,

![Figure 7.2. Edge operator fails in detecting break-line](image)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
as shown in Figure 7.2. It shows an area where the Canny edge operator fails to detect a break-line, the top ridge line of the building, due to a low contrast in that area.

It is possible to recover a break-line by intersection of two existing planes. The concept is depicted in the Figure 7.3.

Points P1 and P2 are on the intersecting line. To determine coordinates of P1 and P2, the x coordinates of the two end points must be given. Then the y coordinates, that
are associated with the given x coordinates, are determined by the line equation:

\[ y = \frac{(A_2 - A_1)x + C_1 - C_2}{(B_2 - B_1)} \]

The above equation is to describe the intersecting line at a constant z coordinate.

Once the x and y coordinates of the two end points are determined, they are used to
determine the z coordinates for each point, using one of the given plane equations. A
numerical example is given in Figure 7.4.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
All computations, shown in the Figure 7.4, use a distance tolerance of 18 centimeters, which is about the accuracy of points in 3D space. By visual investigation, it is seen that the break-line is projected back correctly on to the original image.

7.4 Recover plane boundary from lines in one of the images

Due to difference in illumination, extracting of linear features of the same objects in different images, for example, from a stereopair, can result in different sets of linear features. In other words, linear features that are extracted from the left image may not show up and be extracted in the right image. In this case, the matching algorithm will fail, resulting in an incomplete object boundary in object space. This type of situation is generally found in the study area. An example is given in Figure 7.5.

From the figure, it is seen that the left border of the building on the left image does not show up, due the similarity of the brightness in the roof area and the side wall of the building. On the other hand, the right image has a clear edge for the same border. The side wall is no longer seen in the right image.

In this situation, it is possible to recover the missing plane boundary by way of ray tracing. Given a point on an image, whose exterior orientation parameters are known, a ray is drawn from the point passing through the projection center and intersecting the terrain, which is a plane in this case. The coordinate of the point where the ray intersects the plane is to be determined.

Unlike the back projection problem, which is solved by the collinearity equation, the problem of ray tracing is not straight forward, since the Z coordinate is a function of
Figure 7.5 Missing building edge in one image

X Y coordinates, which are not known at the time of solving. One way to solve the problem is to use an iterative approach, which is depicted in Figure 7.6.

This iterative method has an advantage in that it is not difficult to implement by a computer program. Its algorithm starts from approximating an initial z coordinate, the X
Y coordinates of a potential point on the terrain are computed by the collinearity equation. Then the computed X Y coordinates are used to determine a new Z coordinate. The process is to be repeated over and over, until the change of the terrain coordinate is insignificant.

### 7.5 Extracting more features from image

The concept of extracting more features presented here can be thought of a local processing of previously described procedures. Rather than acting on a global area, it is applied at a particular area of interest. At the beginning, the whole study area is the area
in which a surface reconstruction is performed. Now, the surface is already represented by a number of planes. It is possible to refocus in more details on each individual single plane. Thus, each of them deserves being a new area of interest for surface reconstructing and the whole procedure can be repeated again. More and more details of

Figure 7.7 Extracting more features from images.
surface, if any, will be found and the representation of a surface will get more accurate
and precise. This could be regarded as a type of hierarchical approaches by using details,
as opposed to image resolution.

The concept is implemented on one of the building in the study area, shown in
Figure 7.7. In the figure, the original edge images, left and right (a and b is the figure),
show that the edges of the roof are not complete. Ridge lines are not detected by the edge
operator due to insufficient changes of brightness. As a consequence, the roof of the
building is first interpreted as a flat horizontal plane. The plane, with its determined
boundary, is then treated as a new working area, where the feature extraction procedure is
repeated with new parameters. Shown in subfigures c and d are left and right images of
the building alone, where new edges inside the building are detected. Hence they will be
then matched and reconstructed as linear features in object space. Eventually, the correct
shape of the roof of the building is reconstructed.

7.6 Plane formats

For the sake of convenience, throughout the research, planes and their associated
information are stored by using an array data structure, a matrix-like format where each
plane is stored in a row of a matrix. In reality, to process, analyze and visualize a surface,
or planes, is as important as to create planes. Thus, it is indispensable to utilize available
commercial software in later stages. In order to describe planes in such a way that they
are able to be read and processed by other conventional software, plane data must be
stored in a more conventional format.
After a survey of popular commercial software, suitable to analyze and visualize surfaces, or planes, it is found that there are two types of format that are suitable to describes planes in object space. They are

1. 2D polygon plane format, and
2. Virtual Reality Modeling Language (VRML) plane format.

7.7 2D polygon planes format

Planes that are used to represent a surface have boundaries; therefore, they can be thought of polygons in the object space. It is possible to utilize the polygon format that is widely adopted in the mapping community, more specifically the Geographic Information System (GIS).

In this way, a plane is described by a closed polygon in two-dimension, by using a list of X Y coordinates of the polygon corners. The plane coefficients, A, B and C in the plane equation, are then kept as polygon attributes. The height of any point on a surface, at a particular X Y coordinate, can be determined by firstly identifying the plane to which the point belongs, and then interpolating the Z coordinate from the plane equation.

Most popular GIS software has utilized this technique to store geographic information. In general, information is stored spatially by x y coordinates, and then attributes are added on top of the spatial information. Thus, a 3D coordinate will be represented by an x y coordinate with an attribute storing the z coordinate.

Either vector GIS or raster GIS system can be equally fitted well for this purpose, that is to store planes as one of the information in the system. Figure 7.8 is an example of
planes, represented by 2D polygons, in vector and raster format. The advantage of importing planes into a GIS is the ability to access all GIS analysis tools, which are well developed over many years. Analysis functions related to terrain analysis, for example, water shade area, contouring, volume computation, and so on, can be used immediately with an efficient performance. There are also many GIS programs that are public domain. Another advantage is the integration of other information.

From a geographic information point of view, surface data is another piece of geographic information. It must be integrated with other information to get the most out of the data. In general, terrain surface data have been a vital piece of information in performing many GIS activities. For a purpose of application processing, it is concluded

Figure 7.8 Plots of 2D polygons of planes in vector (Left) and raster (right).
that, using GIS format (2D polygon) to store a surface, or planes, is one of the best solutions to utilize surface data.

7.8 VRML planes format

VRML (Virtual Reality Modeling Language) is a simple text language for describing 3D shapes in an interactive environment. The language is originally designed for the Internet community, in which people can interactively view 3D objects over the Internet. This means the user can interactively control how he wants to see the scene and he will see exactly the objects in the scene as if he were in the real scene.

The advantage of using VRML to describe a surface, or planes, is the ability to visualize objects so that the user is virtually looking at the scene in reality. Functions like walking through, flying through, changing viewing directions, rotating view, and many other functions are made available to the user, so that the user can walk or fly through or around the scene in any way he likes.

VRML is actually specification of a language, not software. To view a scene written in VRML, one must use a VRML viewer, which is basically graphic software to display 3D objects. There are a lot of VRML viewers, sometimes called VRML browsers, available on the Internet and most of them are free software. Thus the user needs only to learn how to describe objects in VRML and it will be ready to be viewed by VRML viewers.
Shown in Figure 7.9 is an example of VRML file. Starting with a declaration of VRML version, the file describes planes by using the index face field. Coordinates of corners of plane are listed first, followed by a sequence of indices in which a surface is formed. In addition, it is possible to add a description of all types of the characteristics of the material used to construct the plane.

It is seen that, in this format, planes are described by a set of 3D points representing polygon corners. In the VRML terminology, polygons are called index faces. Here the plane equation is implicitly stored with the coordinate list. The VRML
viewer software determines how to handle the display of surface, which is described by planes.

According to the last assumption made at the beginning of the chapter, all planes that are disconnected are connected to the ground surface by vertical planes. In this way, the appearance of surface objects look closer to buildings in reality. The VRML viewer used in Figure 7.10 is called “Gl view”, public domain software available at http://home.snafu.de/hg/.

Figure 7.10 Snapshot of VRML viewer of the study area

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Unlike surfaces in GIS, surfaces in VRML are stored in the real 3D coordinate system. They can be processed directly by computer graphic algorithms. Therefore, using VRML format to describe a surface is better in that it is a real 3D surface visualization tool. On the other side, using GIS format to store planes has an advantage of accessing GIS operations as well as data integration.

In both cases, topologies of lines are required so that there is no redundancy in point data. In other words, there will only be one list of points, which contains all coordinates with ID or indices. Then, to construct a polygon, only information of a sequence of point IDs or indices is needed. Using indices is simpler; however, it is weaker in that if there is a change in the point list, the information of polygons must be updated.
CHAPTER 8

DISCUSSION

In this chapter, processes to reconstruct a terrain surface by using 3D linear features are summarized along with a discussion of various points of interest. The advantages of representing a terrain surface by planes, over the two traditional methods DEM and TIN, are discussed. This chapter also discusses the benefits obtained from grouping data in 3D, while knowledge in the 2D image space is from time to time extracted and brought back into the object space. The end of this chapter describes problems that are encountered in this research.

8.1 The big picture

It should be remembered that 3D linear features in the object space do not immediately represent terrain surface. This research is an attempt to reconstruct surfaces by using 3D linear features. The beginning of the process starts in Chapter 5 working all the way to Chapter 7. The flowchart, shown in Figure 7.1, is presented here again to show the whole process of surface reconstruction using 3D linear features.
Figure 8.1 Flowchart of surface reconstruction using 3D linear features
Starting from raw data, 3D poly-lines are segmented into straight lines, then classified into ground and non-ground lines. This step is critical as it requires a good knowledge of the terrain elevation. In this research, where the ground elevation is unknown, a histogram threshold technique is utilized to isolate non-ground lines from ground lines. Next, a ground plane is fitted to the ground line data. The concept behind it is the natural ground surface is most likely a simple and smooth surface, hence there is no break-line. Thus, it is reasonable to fit a surface to a natural terrain. It should be noted that, depending on the terrain type, other higher complex surfaces, for example, a polynomial surface, can be used to represent the ground surface.

According to the data in Table 6.1, the ground plane equation in the study area is generated from 119 lines by least squares adjustment, a weighted mean method. The average residual that observation lines may deviate from the fitted plane is about 0.49 meters. This figure is rather large; however, it is put here for the sake of a discussion. By visual investigation, it is found that there are non-ground lines that are misinterpreted as ground lines. These are mainly lines from cars and man-made objects near the ground, for example, ramps or footpaths. Unless introducing more human knowledge, it is difficult to isolate those near ground lines from the real ground lines.

Non-ground lines refer to lines from manmade objects. In urban areas, like in this research area, non-ground lines are mostly from buildings, and they are most likely from roofs of building. According to the working data, most lines are horizontal lines and low inclined lines.
Parallel lines, horizontal lines and intersection lines are searched exhaustively in the data, within a certain limit of distance. All three cases can lead to a potential plane in the object space, and it will be tested whether it is a physical plane, by associating knowledge from the image. Here is another critical point where human knowledge may be needed in order to get a real physical plane in the object space. However automatic verification of planes is not impossible. In this study a histogram is used to check the homogeneity of the area of a potential plane on the image. However, in some cases where two parallel lines of different buildings are near each other, the automatic plane finding will be fooled by the area between the two lines, which will appear fairly dark, with a homogeneous intensity.

To determine the plane boundary is the last important task, in which, once again, knowledge in the image space is employed. It is found that incomplete plane boundaries are largely caused by the imperfection of edge detection. In many cases a building edge appears only in one of the stereo images; hence reconstruction of lines in the object space is impossible.

It is seen that, throughout the process of surface reconstruction using linear features in object space, knowledge from image space is from time to time extracted and brought back in to object space in order to help answer certain questions. It is very important to be able to extract and fetch information from images quickly and efficiently.
8.2 Advantages of surface representation by planes

Using planes to represent a surface has some advantages over traditional methods. The following is an attempt to compare the method of representing a surface by planes with the two most popular surface representations, Grid Digital Elevation Model (DEM) and Triangle Irregular Network (TIN), in the following issues.

1) Level of abstract description

Planes that represent a surface are considered a higher level of abstraction than both DEM and TIN. Raw data of TIN are randomly distributed over the terrain, while grid elevations in DEM are merely from a densification process. Representing a surface by DEM may be convenient; however it is a loose way to represent a surface. Like an image, DEM is dump data, in which everything is implicit. This makes an interpretation of DEM difficult in terms of extracting information such as break lines, form line, surface roughness.

2) Data volume.

Both DEM and TIN have redundant data. DEM have elevation everywhere at grid posts even if the surface is flat. TIN data, if not well collected, for example, by a laser scanner, can also have a tremendous amount of redundant data. As a consequence both methods need large data storages. On the other hand, a surface represented by a plane does not have redundant data. Planes that are in other planes are already combined at the beginning. Hence this method requires a smaller data storage.
3) Surface breakline.

Both DEM and TIN have an ability to incorporate breaklines into the data set, however, breaklines cannot be explicitly seen in the data set. Rather whenever breaklines are needed, they must be extracted from the data set. On the other hand, surface representation by plane has explicit breaklines because they are in fact plane boundaries.

4) Segmentation.

This is a consequence of having redundant data. Both DEM and TIN need segmentation, which can be considered a type of data cleaning process. Segmentation is always required before other advance data processing, for example, object recognition. However, surface representation by planes does not need segmentation, because the data is already segmented.

5) Height interpolation.

Both DEM and TIN have different approaches in determining the height of a point on the surface. DEM can interpolate a height by using the nearest neighbor, bilinear or bicubic method. TIN must first determine in which particular triangle a point being interpolated falls. Then an interpolation strategy, normally linear interpolation, is used to determine the height of the point. A surface represented by planes must also employ the same approach as that of TIN. It is seen that height interpolation in DEM can be the least accurate, depending on the quality of the interpolated grid DEM. Height
interpolation in TIN is more accurate; however, it will take a longer time due to redundancy of data.

It should be noted that traditional DEM can still be derived from planes, whenever needed, as can contour lines that may be required in the case of producing conventional maps. It is anticipated that the new way of representation of terrain surface by planes and boundaries will enhance analyses and the performance of GIS, mainly because there is no redundant data.

8.3 Advantage of grouping features in 3D

The advantage of grouping features in 3D space is the ability to have one additional piece of information, the height. In fact performing grouping in 3D does not mean that one has to give up all the knowledge in 2D, the image space. Knowledge in the image domain is essentially very important and must be used as well in the object space.

Computer vision systems that extract information in 2D might be fooled by similar orientation of objects, resulting in a wrong grouping of two different objects. One example is linear features of shadows of buildings on the ground. They are likely to be grouped and recognized as buildings. However, by analyzing 3D data, they can be immediately classified as ground lines; hence, they will play no role in building construction.
Another example is shown in Figure 8.2, in which two buildings, A and B in one of the images, are located in the same row right next to each other and may be grouped together into one polygon. In fact, the two buildings have a very distinct height difference, one is 6.5 meters and the other is 8.3 meters. If grouping in 3D, the 2 buildings will be clearly differentiated due to a large difference in height.

It is seen that having 3D data can help recognize things that might be missed by analyzing 2D data. However due to the availability of the relationship between the object

Figure 8.2 Two buildings with different heights
space and the image space, more information can always be extracted and fetched into the object space whenever needed. This is actually the real beauty of photogrammetry, where an area is always covered by at least two photographs, hence enabling the acquisition of 3D information. Figure 8.3 shows a typical overlap in a traditional aerial photography.

![Image]

Figure 8.3 Typical overlap in aerial photography
It is possible to set up a system of 3D object recognition based on the traditional configuration of photogrammetry. A 3D object recognition system leads to an expert photogrammetry work station that is capable of doing object recognition and scene interpretation. The new concept of a digital photogrammetric work station would allow loading all the necessary multiple images of the interesting area into the computer memory. They are to stand by for image query from the 3D object space. Conceptually the system must consist of two major subsystems, one will handle all processing in the image space and the other will handle those in the object space. All modules and functions, either in the image or object space, are actually interrelated and will communicate with each other. When it is requested, information can be extracted at any level of processing. For example, the matching function can request more features from the feature extracting function. Likewise, functions in the object space may require information, such as statistics, from the matching function at any time. By working together in this way, it is expected that automatic object recognition and scene interpretation systems can be made possible.

8.4 Problems

- The most obvious problem encountered in the research is shadows in the image. Shadows in the images play a major role because they show up strongly in the edge image. Shadows on the ground create fewer problems because they will be classified correctly as ground lines. However shadows that are on buildings, once
reconstructed, may have orientation similar to real building edges. Unless using human visual knowledge, the problem is very difficult to solve in automatic mode, and will lead to wrong interpretation of planes.

- Another problem is the inaccuracy in geometric position of 3D linear feature data. This creates a problem since analysis of 3D data, for example, finding parallel lines, intersection lines, plane fitting, completely rely on geometric properties of features. It is found that most inaccurate linear features are from horizontal edges in the images. Horizontal edges in the images are edges that are parallel or closely parallel to the baseline of the image. Great care must be exercised when reconstructing horizontal edges from images. This problem is in fact out of our control and beyond the scope of this research. However, an idea of the accuracy of the 3D data can help processing data correctly, for example, in setting a tolerance value for line intersection.
CHAPTER 9

CONCLUSION

This research focuses on the problem of postprocessing of photogrammetrically reconstructed data after automatic matching using linear features. As previously presented in Chapters one and two, there is a lot of research in photogrammetry and computer vision regarding feature extraction and automatic matching using linear features. The success of this research has made available a data set of linear features in 3D object space. However, on the other side of the story, there is little research that is specifically devoted to handling those linear features in object space. Thus the motivation of this research was set toward using 3D linear features in the object space.

Although linear features in the object space are not the representation of a terrain surface, they carry good information, once extracted, which can be used to generate a complete representation of terrain surface. The fact is that 3D linear features are constructed from linear features in the image space, which are likely to be boundaries of objects in the terrain, hence breaklines. Whether or not those lines in the object space are real physical breaklines can be tested and verified later as parts of the ongoing process.
Therefore it is possible to fit planes to the breaklines and create a complete surface of the terrain.

This research leads to the conclusion that one way to utilize linear features in the object space is to use them to construct a surface, and this is one of the objectives proposed at the beginning of the research. Another objective is to develop a higher level of surface representation that is more suitable for object recognition and scene interpretation than traditional DEM. Both objectives are achieved in this research by introducing a new surface representation using planes and their boundaries. The new representation is superior to the traditional DEM and TIN in terms of higher abstraction of information, no redundancy in the data, and the ability to explicitly describe breaklines in the data. Available popular systems such as GIS can be utilized to handle and maintain the new representation of surfaces without difficulty, resulting in a more efficient technique in terrain analysis.

Another objective of this research is to develop a general grouping scheme to use with 3D linear features. This objective is achieved by introducing a number of segmentation techniques throughout the process in this research. The purpose of these grouping techniques is to make possible the plane finding process, used in Chapter six. Examples are segmentation to straight lines, classification to ground and non-ground lines, parallel line grouping, horizontal line grouping and intersection line grouping.

It is accepted that feature extracting and matching features in images is possible and workable; however, they are still far from the definition of complete and perfect. This leads to the last objective of this research, to incorporate knowledge that is obtained
in the object space into the image space to help balance the imperfection of feature extracting and matching. This objective implies two very essential concepts. First it is not necessary to wait until the problems of feature extracting and matching are completely solved that an analysis of reconstructed features in object space should begin. In other words, it is fine to use such partially perfect existing 3D data to proceed to a higher extraction of knowledge. Second the improvement of feature extracting and matching can be done by not only using knowledge in the image space but also knowledge in the object space. The research has proven that missing lines in the stereo images can even be recovered by using only data in the object space, for example, by plane intersection.

Although there are many buildings detected in the experimental data set, the whole idea of data segmentation in the object space is not just for building recognition. Rather it will be a concept for general object recognition and scene interpretation.

Throughout the research, whenever possible, automatic processing is implemented and utilized as much as possible, for example, line segmentation, classification to ground line, line grouping, and plane fitting. More complex tasks, such as verification of physical planes, verification of plane boundaries, finding missing lines, etc. can be done in automatic mode, however, with less chance of success. Until an expert system can be made, human knowledge is still very important.

Another point to be made is that this research has tried to keep the level of domain knowledge to a minimum. The reason is to have the research focus strongly on the main problem. However, in reality some knowledge might be available, for example, DEM,
laser data, GIS database, etc. This might be able to lessen the pressure of introducing
human interaction with the system. In other words, an automatic system can be achieved
if more domain knowledge is available.

In general all the objectives set at the beginning of the research have been met.
However improvements in various points are definitely essential, since this area of
research is still considered young. They are to be addressed in the next section.

9.1 Future work

This research is an attempt to shift the focus from image processing in 2D space
to data processing in 3D object space. Although this research only concentrates on one
particular task, the surface reconstruction, many subtasks have come up along the way.
There is no way to make all the subtasks and algorithms perfect the first time. The
following is a list of things that can be improved in terms of future work.

1. To find a better criterion to segment ground lines and non-ground lines. This
   might include an adaptive technique, where segmenting parameters can be
   changed over the area, or subdivide the area into smaller pieces before
   segmenting each piece individually,

2. To introduce more intelligent automatic processing, for example, polygon
   searching and boundary finding,

3. To improve efficiency of algorithms in geometric processing. This includes
   algorithms for distance calculation, line intersection, plane intersection, line
   linking, plane fitting and all related photogrammetry functions.
4. To find more alternative ways for testing a plane whether or not it is a physical one. Likewise, alternative solutions to verify plane boundaries must be developed.

9.2 Recommendations

The final goals of digital photogrammetry are much far beyond those of classical photogrammetry. Automatic object recognition and scene interpretation will be a long quest for digital photogrammetry, though not impossible to achieve. Automatic orientation, automatic feature extracting and automatic reconstructions are just beginning steps of the long journey.

In the past, classical photogrammetry had been staying modest in automatic computer vision tasks, for example, object recognition, simply because aerial photographs are available only in the analog format. All interpretation work is done by human visuals. Previously every effort had been put toward the geometric property of photography so that there is as little geometric error as possible in the final product.

Nowadays, things have changed, photography is digital and resides in a computer. This is the main encouraging reason that photogrammetry has focused more on automatic scene interpretation and compilation. Photogrammetry has an advantage of using overlap imagery, which allows access to 3D data at will. The relation between 2D images and 3D object space always exists and is ready to be queried.

In order to step forward, research in photogrammetry should shift toward data processing in the object space. However knowledge in image space cannot be forgotten, for it is where the real raw data reside. The amount of knowledge in the image domain is
considered much more than that in the object space, for it is denser in terms of radiometric quantization and spatial resolution. Likewise knowledge in the object space can also be used to help image processing activities in the image space, for example, feature extraction and feature matching.

From time to time, fetching knowledge from the image space domain is a must and should be done whenever more information needs to be extracted, or a finding in object space needs to be verified. This strategy has been employed throughout this research. This way, it can be made sure that automatic object recognition and map compilation will be possible in the near future. Conceptual details of such an expert digital photogrammetric system are described in Chapter eight.

On the other hand, in computer vision where knowledge is mainly extracted from 2D images, there will be less chance that automatic object recognition and scene interpretation will be completely successful. By success it is meant that the ability of an object recognition system can surpass or at least equal human capability.

It is a fact that human beings have two eyes and use two pictures to interpret objects. Thus, it is extremely important to realize that a system that wants to compete with the human interpretation system must utilize at least two images.

Photogrammetry has always realized and utilized this concept, however, just for data collecting purposes. It is time that photogrammetry exploit the full capability of its own system and knowledge to solve the most difficult problems in surface reconstruction, object recognition and image understanding.
BIBLIOGRAPHY


Li, J. C. and Schenk, T., 1990. Aerial Image Matching Using ψ - s Representation. Notes in Photogrammetry No. 4, Department of Geodetic Science and Surveying, the Ohio State University.


