A NOVEL FEATURE FOR REGISTRATION OF
OBLIQUE AERIAL IMAGES UNDER LARGE
VIEWPOINT VARIATIONS

by

Mao Mao

B.Eng., Beijing Technology and Business University, 2010

a THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in the

School of Engineering Science

Faculty of Applied Sciences

© Mao Mao 2013

SIMON FRASER UNIVERSITY
Summer, 2013

All rights reserved.
However, in accordance with the Copyright Act of Canada, this work may be
reproduced without authorization under the conditions for “Fair Dealing.”
Therefore, limited reproduction of this work for the purposes of private study,
research, criticism, review and news reporting is likely to be in accordance
with the law, particularly if cited appropriately.
APPROVAL

Name: Mao Mao

Degree: Master of Applied Science

Title of Thesis: A Novel Feature for Registration of Oblique Aerial Images under Large Viewpoint Variations

Examining Committee: Dr. Ljiljana Trajkovic
Chair

Dr. Parvaneh Saeedi
Senior Supervisor
Associate Professor

Dr. Bozena Kaminska
Supervisor
Professor

Dr. Shahram Payandeh
Internal Examiner
Professor
Engineering Science

Date Approved: April 25, 2013
Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the “Institutional Repository” link of the SFU Library website (www.lib.sfu.ca) at http://summit/sfu.ca and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author’s written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, British Columbia, Canada

revised Fall 2011
Abstract

This thesis describes a novel image feature, called 2 Edges and one Corner (2EC), that can be used to establish robust match correspondences between aerial images of urban regions under large projective transformation. As implied in its name, the proposed feature is defined by two line segments and their intersection. The lines are constrained to correspond to the boundaries of building rooftops, while the intersections correspond to the rooftop corners. The specific structure of the proposed feature enables us to utilize both geometrical and image context information surrounded by each feature to robustly establish match correspondences between two or more images. To demonstrate the effectiveness of the proposed feature in the matching problem, test dataset images from Pictometry International Corp. have been utilized. Our experimental results show that an average matching accuracy of over 90% can be achieved. These results are superior comparing to the state of the art, which reports an average matching accuracy of 86.32%. 
Acknowledgments

I would like to express my sincere appreciation to my supervisor, Dr. Parvaneh Saeedi, for accepting me as one of her graduate students and providing brilliant guidance during my study and research in her laboratory. Without your enduring support and hard work for providing me with feedback and assistance, this thesis and my successful graduation would not be possible. I am also grateful to my examining committee, Dr. Bozena Kaminska, Dr. Shahram Payandeh, Dr. Ljiljana Trajkovic, for taking precious time out of their schedules to review my work. Lastly, I would also like to thank all my friends and my family for their support and encouragement throughout my academic career.
## Contents

Approval ii  
Partial Copyright License iii  
Abstract iv  
Acknowledgments v  
Contents vi  
List of Tables ix  
List of Figures x  
List of Abbreviations xii  
List of Symbols xiii  

1 Introduction 1  
  1.1 Contributions .................................................. 4  
  1.2 Thesis Organization ............................................. 4  

2 Previous Works 5  
  2.1 Point Matching .................................................. 6  
  2.2 Line Matching ..................................................... 8  
  2.3 Establishing Match Correspondences in Aerial Images .......... 10  
  2.4 Our Proposed Features for Oblique Aerial Images Registration . 14
# Methodology

## 3.1 Introduction

- 3.1.1 Input Data
- 3.1.2 Overview of the Proposed Method

## 3.2 2EC Feature Extraction

- 3.2.1 Straight Line Extraction
- 3.2.2 Line Linking
- 3.2.3 Edge Map Creation
- 3.2.4 Feature Extraction

## 3.3 Locating Feature Correspondences Between Aerial Image Pairs

- 3.3.1 Background of Multiple View Geometry
- 3.3.2 Overview of the Matching Process
- 3.3.3 Feature Location Prediction
- 3.3.4 Search Space Creation
- 3.3.5 Filter Candidate Features Based on Feature Shape
- 3.3.6 Filter Candidate Features Based on Image Correlation
- 3.3.7 Removal of the Outliers

# Experimental Results

## 4.1 2EC Features Evaluation

- 4.1.1 Feature Detection Results
- 4.1.2 Feature Matching Results

## 4.2 2EC Feature Application On Aerial Images Matching

- 4.2.1 Experimental Results
- 4.2.2 Comparison with the State of the Art
- 4.2.3 Discussion of the Experimental Results
- 4.2.4 Limitation of 2EC Features

## 4.3 Discussion of the Time Complexity

# Conclusions

## 5.1 Summary of Contributions

## 5.2 Potential Future Research

# Appendix A Input Image Pairs
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Partitioning Pixels into Gradient Orientation Bins.</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>Partitioning Pixels into Shifted Gradient Orientation Bins.</td>
<td>19</td>
</tr>
<tr>
<td>4.1</td>
<td>Results of match correspondences establishment using 2EC features</td>
<td>54</td>
</tr>
<tr>
<td>4.2</td>
<td>Results of match correspondences establishment using SIFT features</td>
<td>63</td>
</tr>
<tr>
<td>4.3</td>
<td>Results of match correspondences establishment using SURF features</td>
<td>63</td>
</tr>
<tr>
<td>4.4</td>
<td>Results of match correspondences establishment using ASIFT features</td>
<td>64</td>
</tr>
<tr>
<td>B.1</td>
<td>Parameters for the line detection and linking algorithms (Sections 3.2.1 and 3.2.2).</td>
<td>78</td>
</tr>
<tr>
<td>B.2</td>
<td>Parameter for edge map creation (Section 3.2.3).</td>
<td>79</td>
</tr>
<tr>
<td>B.3</td>
<td>Parameters for 2EC feature extraction I (Section 3.2.4).</td>
<td>79</td>
</tr>
<tr>
<td>B.4</td>
<td>Parameters for 2EC feature extraction II (Section 3.2.4).</td>
<td>80</td>
</tr>
<tr>
<td>B.5</td>
<td>Parameters for search space creation (Section 3.3.4).</td>
<td>81</td>
</tr>
<tr>
<td>B.6</td>
<td>Parameters for filtering candidate features based on feature shape (Section 3.3.5).</td>
<td>81</td>
</tr>
<tr>
<td>B.7</td>
<td>Parameters for filtering candidate features based on image correlation (Section 3.3.6).</td>
<td>81</td>
</tr>
<tr>
<td>B.8</td>
<td>Parameters for removal of the outliers (Section 3.3.7).</td>
<td>82</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Nadir and oblique looking cameras .......................... 3

3.1 Example of oblique aerial image pair .......................... 16
3.2 Example of metadata file .......................... 16
3.3 System flowchart .......................... 17
3.4 Feature extraction flowchart .......................... 18
3.5 Conversion from gradient orientation bins to line support regions .......................... 20
3.6 Conversion from line support regions to lines .......................... 20
3.7 Four criteria for line linking process .......................... 21
3.8 Example of detected line segments on the edge map .......................... 23
3.9 Example of line merge .......................... 24
3.10 Example of determining endpoints for line segments .......................... 25
3.11 Line removal criteria .......................... 26
3.12 Illustration of feature extraction .......................... 27
3.13 Results of feature extraction .......................... 28
3.14 Feature prediction procedures .......................... 34
3.15 Example of feature prediction .......................... 36
3.16 Histogram for the predicted feature location offsets .......................... 37
3.17 Histogram for the predicted epipolar line accuracy .......................... 37
3.18 Histogram for the predicted line segments length difference .......................... 37
3.19 Histogram for the predicted line segments orientation difference .......................... 38
3.20 Search space .......................... 39
3.21 Example of 2EC feature matching based on correlation .......................... 43
3.22 Outliers removal flowchart .......................... 44
3.23 A pair of sample images with all the matches identified .......................... 45
4.1 Results of 2EC features detection on small images ........................................ 48
4.2 Results of SIFT features detection on small images ........................................ 49
4.3 Results of 2EC feature based matching for small images ............................... 51
4.4 Results of projective and affine transformations removal for small images ... 52
4.5 Results of SIFT feature based matching for small images ............................... 52
4.6 Results of matching in aerial image pair using 2EC features ............................ 56
4.7 Box filters ........................................................................................................... 58
4.8 Aerial image pair after removing the projective and affine transformations
and viewpoint rotation .............................................................................................. 61
4.9 Comparison of matching results using 2EC and SIFT features ................. 65

A.1 The matching result of image pair 90_1. ......................................................... 72
A.2 The matching result of image pair 90_2. ......................................................... 72
A.3 The matching result of image pair 90_3. ......................................................... 73
A.4 The matching result of image pair 90_4. ......................................................... 73
A.5 The matching result of image pair 90_5. ......................................................... 73
A.6 The matching result of image pair 180_1. ......................................................... 73
A.7 The matching result of image pair 180_2. ......................................................... 74
A.8 The matching result of image pair 180_3. ......................................................... 74
A.9 The matching result of image pair 180_4. ......................................................... 74
A.10 The matching result of image pair 180_5. ......................................................... 74
A.11 The matching result of image pair 270_1. ......................................................... 75
A.12 The matching result of image pair 270_2. ......................................................... 75
A.13 The matching result of image pair 270_3. ......................................................... 75
A.14 The matching result of image pair 270_4. ......................................................... 75
A.15 The matching result of image pair 270_5. ......................................................... 76
List of Abbreviations

ASIFT  Affine Scale Invariant Feature Transform
DoG    Difference of Gaussian
EBR    Edge Based Region detector
ED-MSER MSER with information Entropy and spatial Dispersion constraints
GTM    Graph Transformation Matching
GSD    Ground Sample Distance
IBR    Intensity extrema Based Region detector
IC     Initial Correspondences
LICF   Line Intersection Context Feature
LiDAR  Light Detection And Ranging
MSER   Maximally Stable Extremal Regions
MSLD   Mean Standard deviation Line Descriptor
NCC    Normalized Cross Correlation
ORSA   Optimized Random Sampling Algorithm
RANSAC RANdom SAmple Consensus
SIFT   Scale Invariant Feature Transform
SURF   Speeded Up Robust Features
2EC    2 Edges and one Corner
2DOC   2D Orthogonal Corners
3CS    3 Connected Segments
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>The first image in an image pair</td>
</tr>
<tr>
<td>$I_2$</td>
<td>The second image in an image pair</td>
</tr>
<tr>
<td>$t_{lsr}$</td>
<td>Minimum area allowed to form a line support region</td>
</tr>
<tr>
<td>$d_l$</td>
<td>Lateral distance threshold in the line linking process</td>
</tr>
<tr>
<td>$t_a$</td>
<td>Angle threshold in the line linking process</td>
</tr>
<tr>
<td>$t_{overlap}$</td>
<td>Maximum overlap ratio allowed in the line linking process</td>
</tr>
<tr>
<td>$t_{underlap}$</td>
<td>Maximum underlap ratio allowed in the line linking process</td>
</tr>
<tr>
<td>$t_{iso}$</td>
<td>Size of the isolated blob on the edge map</td>
</tr>
<tr>
<td>$t_{pline}$</td>
<td>Overlap ratio of lines with the edge map</td>
</tr>
<tr>
<td>$l_{E_{max}}$</td>
<td>Maximum length of extension allowed for each line</td>
</tr>
<tr>
<td>$l_{E_{min}}$</td>
<td>Maximum length of continued extension for each line</td>
</tr>
<tr>
<td>$t_{ma}$</td>
<td>Angle threshold in the line merge process</td>
</tr>
<tr>
<td>$d_{ml}$</td>
<td>Lateral distance threshold in the line merge process</td>
</tr>
<tr>
<td>$\theta_{min}$</td>
<td>Minimum cross angle for the two lines to be intersected</td>
</tr>
<tr>
<td>$\theta_{max}$</td>
<td>Maximum cross angle for the two lines to be intersected</td>
</tr>
<tr>
<td>$t_{closeinter}$</td>
<td>Line length threshold for removing single short lines</td>
</tr>
<tr>
<td>$t_{short}$</td>
<td>Line length threshold for removing short lines that are connected together forming a chain</td>
</tr>
<tr>
<td>$t_{fex}$</td>
<td>Minimum line length to extract 2EC features</td>
</tr>
<tr>
<td>$P_{abc}, Q_{abc}$</td>
<td>2EC features in $I_1$ and $I_2$</td>
</tr>
<tr>
<td>$P_{a'b'c'}$</td>
<td>Transformed 2EC feature from $I_1$ to $I_2$</td>
</tr>
</tbody>
</table>
The centers of the two cameras

Point in the 3D world

Point in the two image planes

Axes of the world coordinate system

yaw, pitch, roll

yaw, pitch, roll angles of the camera

longitude, latitude, and altitude of the camera

Camera matrices of the two cameras

Camera calibration matrices of the two cameras

Rotations from the world coordinate system to the cameras

Translations from the world coordinate system to the cameras

Focal length of the two cameras

The coordinates of the principal points in the two input images

The distance between the pixels in the metric world in the x and y directions

3D coordinates of the two cameras with respect to Earth-Centered, Earth-Fixed coordinate system

Dimension of the input image

Projective matrix that relates the two input images

End points of a 2EC feature

Center point of a 2EC feature

Size of the square neighborhood around the predicted location in $I_2$

Width of the stripe neighborhood along the epipolar line in $I_2$

Search space in $I_2$ for $P_{abc}$

Length of the two line segments in a 2EC feature

Orientation of the two line segments in a 2EC feature
\(T_{len}\) Maximum length difference allowed for two 2EC features to become a match

\(T_{angle}\) Maximum orientation difference allowed for two 2EC features to become a match

\(I_x, I_y\) Gradient magnitude of the image in the horizontal and vertical directions

\(\nabla I_1, \nabla I_2\) Gradient images

\(f_s\) Sobel operator

\(x_k\) Pixels inside the image region surrounded by the 2EC feature in \(I_1\)

\(K\) Number of pixels in the image region surrounded by the 2EC feature

\(x'_k\) Transformed pixels in \(I_2\)

\(\mathbf{vt}\) Translation vector

\(\nabla I_{P_{abc}}, \nabla I_{Q_{abc}}\) Average gradient values inside the image patches surrounded by \(P_{abc}\) and \(Q_{abc}\)

\(D_{PQ}\) Correlation score

\(t_{corr}\) Correlation score threshold used to determine the feature correspondences

\(p_i, q_i\) Putative point match correspondence

\(\mathbf{H}\) Homography transformation estimated using the initial set of point correspondences

\(s_i\) Residual error for each potential match correspondence

\(\bar{s}\) Average residual error

\(\sigma(s)\) Standard deviation of residual error

\(f_{ac}\) Number of iterations

\(t_{fac}\) Factor in the iteration process

\(s_{tre}\) Maximum residual error allowed to stop the outliers removal iteration
\( n \)  
Minimum number of match correspondences allowed to stop the outliers removal iteration

\( s_{\text{larg}} \)  
Maximum residual error

\( \mathbf{H}' \)  
Updated homography transformation

\( l_i, l'_i \)  
Epipolar lines

\( F \)  
Fundamental matrix

\( c p_i, c p'_i \)  
Ground truth correspondence

\( \text{Err} \)  
Average distance from the estimated epipolar lines to the ground truth correspondences

\( T_1, T_2 \)  
Transformations that remove the projective and affine transformations from the original images

\( I'_1, I'_2 \)  
Images after the projective and affine transformations removal and viewpoint rotation
Chapter 1

Introduction

The objective of this thesis is to design robust and novel features that can be robustly and automatically matched between oblique aerial images of scenes with large view variations. Many methods were suggested during the last several decades [1] for reconstructing 3D scenes using 2D images of those scenes. Most of these methods, however, require establishing match correspondences in different views to compute the camera transformation parameters between the two views and then reconstruct the 3D representation of each scene using stereo algorithm. Therefore, automatically establishing correct match correspondences between two views of a scene is a requirement for most 3D reconstruction algorithms that rely on multiple 2D views. In most of the above methods, the difference between the viewings of the scene is small, and therefore establishing match correspondences between two scenes is not as challenging as for cases where there is a significant variation between the two views of the same scene. This is due to the fact that local characteristics of features could change dramatically when viewed from substantially different views. In this thesis, we are interested in defining and establishing match correspondences for aerial images that include large view variations.

Most of the existing matching algorithms that are designed for 3D reconstruction could only cope with smaller view variations. This means that the two scene images are related by small transformations that usually include translation, rotation, and small scale changes. The main reason for such bias is that the similarity of regional neighborhood of the salient image features can provide acceptable results. However, since the capturing cameras are close to each other, comparing to the distance of the scene to the cameras, the estimation of the depth may include large errors and uncertainties when the 3D reconstruction task
is performed [2]. Moreover, a complete 3D reconstruction of a scene requires processing multiple views of a scene from different views that overall completes 360 degrees view of that scene. In some applications, acquisition of scene images with 360 degrees of view coverage is possible, but for most others, especially those used for 3D reconstruction of large outdoor scenes, it is not possible. For instance, reconstruction of 3D model of a building requires access to all side views of that building, which results in requiring tremendous amount of data and processing resources. A solution to this is the use of wide baseline images for the 3D reconstruction problem. Such images are captured from the viewpoints that are considerably apart, but cover a larger view of the scene. For example, a camera may capture only 4 images, each with 90 degrees difference from the previous view. We refer to such images as images of wide baseline. Reconstruction using less number of images requires less computing resources and processing time, and therefore advantageous to those methods that require large number of images. Reconstruction using wide baseline images is a very challenging problem due to the fact that when viewpoints are too far apart, identical features of a scene often include tremendous visual differences (as they are viewed from different perspective) and therefore hard to be associated with each other using automatic approaches.

The problem of establishing match correspondences between wide baseline images has been the subject of interest in the image processing and the computer vision fields for several decades. Many researchers have tried to address this problem using different techniques and image resources. Some limited successes are reported in [3, 4, 5]. In [3], wide baseline matching was achieved by extracting Maximally Stable Extremal Regions (MSER) based on intensity thresholds. MSER was reported in [6] as the most stable affine invariant region feature compared to the other features including Harris-Affine features [7], Hessian-Affine features [8], Edge Based Regions (EBR) [9], Intensity Based Regions (IBR) [10] and salient regions [11]. However, as pointed out in [6], the repeatability for these features decreases as the viewpoint variation increases. In [4], Wang et al. proposed a line segment clustering algorithm for matching low-textured wide baseline images. This algorithm claimed to be able to tolerate large viewpoint changes, and perform better than local features in matching low-textured images. Nonetheless, the geometrical traits of the line groups could not hold constant under extreme viewpoint variations. In [5], Morel and Yu extended the idea of SIFT and developed a fully affine invariant matching algorithm, Affin-SIFT (ASIFT). However, the high computational cost became its biggest issue, although a two-resolution procedure
was proposed to accelerate ASIFT. The complexity of ASIFT was 2.25 times as much as that of SIFT [5], which hindered its application on images of large sizes. Besides, ASIFT failed in matching repetitive patterns or low-textured objects, which were very common in man-made scenes.

In this thesis, we address the problem of robust feature identification and matching in oblique aerial images of urban areas with large baselines. The term oblique refers to the fact that the capturing camera plane is at an angle with respect to the ground surface. Figure 1.1 shows an example of a nadir looking camera and an oblique looking one.

![Figure 1.1: Nadir and oblique looking cameras](image)

Clearly, scene images taken at wide baselines with oblique view cameras go through a substantial visual change that makes the application of the previously suggested algorithms on these images hard if not impossible. For instance, in scene subjects with more texture and objects, the majority of the previous works suggest using interest points and their local attributes to establish match correspondences. Such features and their local image regions undergo large amount of change when viewed at large baselines. For less textured scenes (such as indoor environment), some methods suggest using edges or lines as features. Edges by themselves could be problematic in large baseline views as their length changes (in an extreme case can be converted into points under 90 degree view variation). Therefore, there is no general algorithm that has fully addressed the problem of wide baseline matching. Those algorithms that succeed in certain type of scenes may fail in others. Besides, although it is claimed that some of those methods are invariant to viewpoint changes, the repeatability of most of their features for wide baseline matching decreases as the viewpoint change increases. Therefore, an algorithm that especially aims for matching correspondences in oblique aerial images with large viewpoint variation is very desirable.

In this work, we propose a novel feature that is specifically designed for matching aerial
images of large baseline with oblique views of urban scenes. The proposed method takes advantage of line segments that naturally exist in the construction of urban environment. It uses line segments as a medium to accurately establish point correspondences. The proposed feature detector starts by detecting lines, and then obtains line intersections. Certain rules are followed to remove spurious lines. The proposed features, which depict the attribute of the man-made structure in urban environments, are then extracted using a permutation method. The proposed features are extracted in input oblique aerial images separately and are later used to establish match correspondences using a novel method. The established match correspondences can be used to compute accurate camera parameters and fundamental matrix and lead to accurate image registration and 3D reconstruction.

1.1 Contributions

In this work, two contributions have been made:

1. We propose 2EC features that hold geometrical traits of straight edges and their intersections with each other and are especially designed to represent typical configurations in man-made structures. Since 2EC features preserve geometrical relationships between object edges, they can be associated with each other even under large projective transformations that may make them look different in different views.

2. A new matching algorithm is developed that utilizes both visual and geometrical properties in a hierarchical way to robustly and accurately establish match correspondences between two or more images of a scene.

1.2 Thesis Organization

The remainder of this thesis is divided into four chapters. Chapter 2 presents the previous works in image features that are utilized for matching purposes. Chapter 3 details the proposed 2EC features and our developed method for matching them in two or more views. Experimental results are presented in Chapter 4 followed by the conclusions and the discussion of the potential future work in Chapter 5.
Chapter 2

Previous Works

3D building reconstruction via multiple images has received increasing interest with the improved imaging technology for capturing high resolution images. Compared to the 3D building reconstruction approaches that use stereo image pairs, methods using multiple images can obtain higher accuracy [1]. The first task in 3D reconstruction is to acquire accurate fundamental matrix by establishing correspondences between image pairs or using precise acquisition geometry. However, it is very hard to have such precise acquisition geometry. Therefore most approaches in this area have focused on establishing correspondences to obtain fundamental matrix. These methods require the extraction of viewpoint invariant regions centered at interest points, within which the local descriptions of the features are estimated. In low-textured environment, however, such points are difficult to detect and local feature descriptors might not be distinctive enough. Due to the high demands for matching in low-textured scenes (such as buildings and walls in man-made environments), line segments have been employed as features by some methods. Among those algorithms, some use lines as a medium to obtain more accurate point matching results by intersecting them, while some match lines directly based on image information along the lines or their geometric properties. More recently, matching techniques have been applied on aerial images so that 3D city modeling can be achieved using two or more 2D images [12, 13, 14, 15]. However, most of these algorithms in this area can only cope with small affine transformations.

In this section, we review the previous works that are developed for the purpose of matching under wide baseline. The use of features is first reviewed followed by the presentation of matching algorithms. Finally, the state of the art for the application of image matching in the domain of aerial imagery matching is reviewed.
2.1 Point Matching

A wide range of approaches involve detection of distinctive points and viewpoint invariant descriptors. The frameworks of these approaches are similar to each other. In summary, in such algorithms, interest points are first extracted from both images independently. Then, a descriptor is computed that uniquely describes each feature point. The descriptor measures unique characteristics inside a region around each feature point, which is invariant to a certain class of transformation. The measurement could be based on the information such as intensity, color, or texture. Next, putative match correspondences are established by comparing the feature descriptors. Finally, incorrect matches are rejected by verifying the geometric consistency of the locations of the putative matches using methods such as GTM [16] or RANSAC [17].

The main difference between each matching algorithm is in how the feature points and their viewpoint invariant regions are defined. Scale Invariant Feature Transform (SIFT), proposed by Lowe [18], defined feature points in scale space using Difference-of-Gaussian (DoG) functions. Based on local image gradients, a SIFT descriptor was computed within a circular region around each feature point. The point correspondences were determined by comparing the descriptor vectors in the Euclidean space. This method was claimed to be invariant with respect to image scaling and rotation. To provide rotation invariant, the descriptor was compared to the orientation assigned to each feature point. This method, however, only tolerated small affine transformations. The matching accuracy was only about 50% when there was a 50 degrees viewpoint variation between the images.

Speeded Up Robust Features (SURF) [19] were introduced to be also scale and rotation invariant. The main focus for these features was on increasing the computation efficiency of feature extraction and matching. SURFs were claimed to be more robust and several times faster than SIFT features. However, SURF was also variant to the viewpoint changes.

In order to cope with wider viewpoint changes, several feature matching methods were proposed based on affine invariance [10, 3, 20, 8, 5]. Tuytelaars et al. [10] proposed to extract affine invariant regions around anchor points. In their work, anchor points were extracted based on local extrema in intensity images. By checking the intensity profile along each ray radiating from the extrema, the maximum point on each ray was selected. These discrete points were then connected together to form a region which was fitted into an ellipse and described with generalized color moments [21]. Both the ellipse region and the descriptor...
were claimed to be affine invariant. In [3], Matas proposed Maximally Stable Extremal Region (MSER) features based on intensity thresholds. To achieve a balance between the distinctiveness and planarity of the features, multiple scales of the MSER regions were selected as feature description regions, within which descriptors were computed based on complex moments. A voting strategy followed by a correlation was employed to determine the true correspondences. Tell and Carlsson [20] proposed a new local feature, which was a line segment that connected to two Harris [22] points. Affine invariant Fourier coefficients were then computed from the intensity profile along the line. Cyclically order constraint was added to achieve better results. This method, however, required textured objects, which were locally planar. Harris affine [8] was proposed by Mikolajczyk and Schmid as an extension of Harris-Laplace [7]. Harris-Laplace defined interest points by multi-scale Harris corner detector. For each interest point, a characteristic scale was then selected, for which the Laplace operator achieved its maximum value. A scale invariant region for each point was determined according to its scale characteristic. Years later, Mikolajczyk and Schmid improved the scale invariant features and made them affine invariant. In [8], elliptical affine invariant regions were estimated using an iterative approach, which was first proposed by Lindeberg [23].

Although claimed to be affine invariant, the repeatability of all the above feature points and their viewpoint invariant regions decreases severely as the viewpoint variation increases between the two images [24]. However, in order to extract viewpoint invariant descriptors, same viewpoint invariant regions around corresponding feature points in both images must be detected, which is very difficult under extreme viewpoint changes. To overcome this challenge, Yu proposed an affine invariant image comparison method, Affine-SIFT (ASIFT), in 2009 [5]. This method was claimed to be fully affine invariant. It generated a set of sample images, which simulated all possible affine transformations of the input images by varying the orientation of the camera's optical axis. SIFT method was then applied to all the sample images to obtain putative matches. Finally, the Optimized Random Sample Algorithm (ORSA) [25] was used to eliminate outliers according to the epipolar geometry constraint. It was shown in [5] that in the case of extreme affine transformation, ASIFT successfully established correspondences, while SIFT [18], MSER [3] and Harris-affine [8] failed. However, ASIFT failed in matching repetitive patterns or low-textured objects, which were very common in man-made scenes. Besides, as we found out in our experiments, although ASIFT could result in large number of good matches, the ratio between the number of correct
matches and the number of total matches sometimes might be low. Therefore, relying on the matching results of ASIFT to compute fundamental matrix might be problematic.

All of the above algorithms assume that the local neighborhood around each feature is planar. This is true only in planar scenes. For example, in scenes containing objects at different distances from the camera, if a point is detected on the corner of an object, its local neighborhood would have depth discontinuities, no matter how the affine invariant region is extracted. In such cases, it is impossible for the feature descriptors (computed via one of the above mentioned methods) to be invariant with respect to an affine transformation. Besides, since affine invariant regions are kept small to satisfy planarity conditions, they may become less discriminative, when there are low-textured objects in the images.

2.2 Line Matching

Above mentioned point matching methods usually fail if the scene contains low-textured objects. However, such scenes usually contain plenty of line segments. Complex object boundaries can usually be approximated with line segments [4]. In such cases, line segments are probably better features to be used. In [26], line matching algorithms are divided into two groups: those that match individual line segments, and those that match groups of line segments. Algorithms reported in [27, 28, 29] (that match individual lines) focus on geometrical traits, such as length and orientation, or the image information in the stripe neighborhood along each line. Other algorithms such as those in [2, 4, 30, 31] match group of lines using the topological relationships of neighboring line segments. Here, topological relationship refers to the relative positions and orientations of local groups of lines [2]. In the second category of algorithms, lines are grouped based on their spatial proximity and matched according to their geometrical attributes that are invariant with respect to viewpoint variations.

In [27], Schmid and Zisserman established line correspondences by computing the cross-correlation score of the intensity of line segments neighborhood. The fundamental matrix was used to generate a point-wise correspondence between potential line matches. The application of this method, however, was restricted since it required a prior and precise knowledge of the fundamental matrix. Bay et al. [28] introduced an algorithm which combined the appearances of lines and their topological relations. Putative line matches were obtained by comparing the color histograms along both sides of each line. The histogram
similarity was measured using Euclidean distance. Next, a topological filter was applied to reject incorrect matches. More matches were then found by iteratively adding line segments which do not violate the current topological structure of the matches. However, the establishment of the initial matches relied on the color distribution of the image along both sides of the line segments, which in general is variant with respect to illumination changes. In [29], individual lines were matched using a novel line descriptor called Mean Standard Deviation Line Descriptor (MSLD). For each line segment, a pixel support region was defined and was partitioned into sub-regions. Each sub-region was then described by a feature vector based on the image gradient values. MSLD was built by incorporating the mean and standard deviation of all the sub-region descriptors. MSLD was claimed to be distinctive for line matching under rotation, illumination and viewpoint variations. It, however, required high repeatability of line segments in scene images.

Tell [2] presented a line matching method by exploring the geometric structures of line segment pairs. The relative position of each pair was represented by an integer index. A voting technique was utilized to find corresponding line pairs with the same index between two images. This method could tolerate quite a large viewpoint variation. However, its correct matching rate was severely influenced by the low repeatability of the line detection results. Besides, when the view variation was large, the invariance properties of the relative positions between line pairs may not have been preserved any longer. Wang et al. [4] introduced a line segment clustering method based on spatial proximity and relative saliency. He defined the set of line segments which were grouped together as a novel feature called line signature. A codebook technology was then applied to fast match line signatures. Line segment matches were finally obtained by a consistency checking method. It was claimed that this method could cope with low-textured images and non-planar scenes under large viewpoint variations. However, the experimental results showed that when the viewpoint variation increased from 10 degrees to 50 degrees, the number of correct matches dropped by 75%. Besides, it suffered from high computational cost when analyzing topological relations. In order to establish correspondences between images of building facades, Lee [30] proposed to detect quadrilaterals which were formed by four line segments and their intersections. A projective transformation model was computed and its accuracy was evaluated by Harris points inside the quadrilateral. This method established substantial number of true correspondences between two images of building facades, even under 50 degrees of viewing angle differences. However, its dependence on a dominant plane, i.e., a building facade, limited
its applications. In order to avoid heavy computation in analyzing the topology of a group of lines and the ambiguity of single line matching, Kim and Lee investigated the matching of line pairs in [31]. They proposed a new feature called Line Intersection Context Feature (LICF). Line intersections were extracted by intersecting pairs of lines based on their spatial proximity. Next, a local region centered at each intersection location was extracted as a LICF feature. To obtain initial corresponding LICF features between two images, normalized cross-correlation (NCC) was utilized. The classic RANSAC method was finally used to eliminate outliers.

Even though line segments have obvious advantages for matching in man-made scenes, they have their own challenges. First, as suggested in [2], their repeatability could be highly affected when two images have large viewpoint differences. For example, a line might be complete in one image, but only partially could be detected in the other. Also, under large projective transformations, topological relationship between groups of lines may not be preserved. Besides, since line segments usually lie on the intersections of two surfaces, the neighborhood appearance of a line segment might change completely if viewed from the different sides of it. If these shortages of line segments can be overcome or avoided, matching man-made scenes using line segments may tolerate even larger viewpoint variation than the above mentioned methods.

2.3 Establishing Match Correspondences in Aerial Images

In recent years, the increased availability of high-resolution aerial photography has made the 3D reconstruction of buildings and scenes using multiple view images more popular than ever. Since oblique aerial images are taken at an angle, not only do they provide information about building rooftops and the ground surface, but they include detailed building facades. However, the automatic oblique aerial image matching is plagued by issues like large projective distortions and occlusions. Oblique aerial image matching for 3D building reconstruction is a subset of the above mentioned wide baseline image matching problem. Although some limited solutions for wide baseline matching are suggested for some specific applications, most of the existing matching algorithms in the domain of aerial imagery can only cope with nadir looking images and the transformations that they can deal only includes rotations, translations and scaling. However, matching oblique aerial images under large projective transformations have not been solved. Here we present a brief review of
CHAPTER 2. PREVIOUS WORKS

the matching algorithms that were developed especially for aerial or satellite imagery. A large number of algorithms [12, 13, 14, 32, 33, 15] detected local features on aerial images or satellite images to solve the matching problem. However, in the realm of 3D reconstruction, line segments are more representative features [34, 35, 36, 37] because they are more present on man-made structures. Unlike line matching algorithms that were mentioned in Section 2.2, these algorithms did not necessarily solve the image matching problem. They either mapped aerial images onto a 3D LiDAR model or matched line segments with a priori known epipolar geometry. Nonetheless, they are reviewed here to demonstrate the use of line segments as features for establishing match correspondences in aerial/satellite images of urban scenes.

Zheng and Chellapa [12] described a general ground plane image registration algorithm to register two aerial images of oblique views. The camera orientations were estimated by an illumination direction estimator [38]. Feature points were extracted based on Gabor wavelet decomposition [39] and matched using an area based correlation function. A projective transformation model was then computed using the putative matches and false matches with high residual error were rejected. Yasein and Agathoklis [13] described a feature-based algorithm for aerial image matching. In their work, feature points were first extracted in both images based on scale-interaction of Mexican-hat wavelets [40]. Next, putative correspondences were found by comparing the Zernike moments-based descriptors [41], which were used to describe the local circular neighborhood centered at each feature point. The transformation parameters, which could be used to register two aerial images, were estimated using an iterative weighted least squared error minimization. Xiong and Zhang [14] proposed a novel interest point matching algorithm for high-resolution satellite images. In their work, networks of control points were formed by a point clustering procedure. The relative positions and orientations between the center point and the end points of the network were used to determine the match correspondences. In [32], SIFT features were detected on remotely-sensed images for registration purposes. It was shown that SIFT features could achieve fairly high matching results (accuracy > 80%) when the two input images were taken from similar directions. Cheng et al. [33, 15] proposed a new affine invariant feature, called MSER with information Entropy and spatial Dispersion constraints (ED-MSER), for remote sensing images registration. In this work, initial features were extracted using MSER [3]. Qualified features were then selected based on information entropy and spatial dispersion
CHAPTER 2. PREVIOUS WORKS

constraints. SIFT descriptors were also utilized to characterize each feature. Initial correspondences were established by computing the descriptor distances in the Euclidean space. Later, in order to remove outliers and obtain optimized matching results, an improved RANSAC method was proposed, which could automatically select the distance threshold for RANSAC algorithm. Since this method repeated RANSAC routine several times, it was very time consuming. Moreover, since the improved RANSAC was based on the assumption that the two image planes were related by a homogeneous transformation, if the height variances between the ground and structures in the scene were large, the optimization results would be affected.

Apart from local features, some previous works in the literature focused on matching aerial images based on lines. In [34], line segments were grouped together to form junctions, parallels and U-contours in a hierarchy way. The grouped features defined hypotheses for flat and gable roofs boundaries and were matched hierarchically based on a quadrilateral constraint, which was generated using the epipolar geometry to reduce the search space. In [35], Ding et al. proposed a method to project oblique aerial images onto a 3D LiDAR model using 2D Orthogonal Corners (2DOC). The features in aerial images were extracted based on the spatial proximity of the line segments and the orthogonal vanishing point pairs. True 2DOC features could present the orthogonal corners of building profiles. Putative correspondences between the aerial image and the LiDAR data were obtained by comparing the spatial proximity and orientations of 2DOC features. A method combining Hough transform and RANSAC was then used to reject incorrect matches. In this work, the number of true 2DOC was significantly influenced by the type of buildings in the scene. For large and simple building shapes such as those in downtown district, the method worked pretty well. However, for complex building structures like those in residential suburban areas, it failed up to 50% of the times. In [36], Wang et al. reported a novel feature called 3 Connected Segments (3CS) to map oblique aerial images onto 3D LiDAR data. Each 3CS feature contained three line segments, which were connected to each other based on their endpoints proximity. Six attributes including segments' length ratios and orientations were used to describe each 3CS feature. Finally, a two-level RANSAC algorithm was developed to remove outliers in those cases where the number of outliers was much larger than the number of inliers. The distinctiveness of 3CS features made the number of correct correspondences among the initial candidates to increase. However, when registering two aerial images, 3CS features could not be used directly, since the repeatability of such features in both images
is usually very low. In [37], a new method for matching line segments in stereo aerial images was presented. To establish line correspondences between two aerial images, line pairs in one image were selected, and their potential corresponding candidates in the other image were identified with the help of a priori known epipolar geometry. The final pairs of matches were determined by comparing a weighted pair-wise matching similarity score, which required information of epipolar geometry as well as the geometric and image context constraints. Finally, the best line-to-line correspondences were determined using a new line-to-line measure based on a voting scheme. It was shown that this approach could achieve promising results with an average correctness rate of 95.2%. However, as explained in the paper, this method would lose its efficiency for a relatively coarse ground sample distances (0.5 – 1.0 meter per pixel). Moreover, in assessing the proposed method, only images with small projective transformation were tested. The efficiency of the method for aerial images with large projective transformation was not discussed or shown.

Besides, for aerial images captured by cameras that are mounted under the aircraft, the distance from the cameras to the scene is high enough such that the height variations of the terrain and the objects on the ground can be ignored. Therefore, the two input images can be related by a homography transformation. Many methods designed for registration of aerial images have utilized this assumption to provide additional constraints when establishing match correspondences. In [15], Cheng et al. removed the outliers based on a homography model. They also pointed out that when the elevation between surface features is large, the power of the homography will decrease, and further affect the accuracy of matching optimization [15]. In [42], affine transformation model was used when feature points were matched using a graph-based matching algorithm. According to this work, the affine transformation model was an acceptable one since the aerial images had only narrow field of views.

However, the above algorithms only provide solutions for aerial images matching under planar rotation, translation or small affine transformations and indeed oblique aerial images with large projective transformations cannot be registered using any of the above algorithms.
2.4 Our Proposed Features for Oblique Aerial Images Registration

In our work, we propose a novel feature, which is later used for automatic match correspondence establishment over two or more oblique aerial images with large projective transformations. As described in previous sections, local feature points are usually not distinctive enough for matching aerial images, which include a large number of low-textured objects. Both lines and corners are used in the design of the proposed complex features as each has its own unique characteristics and shortcomings.

Instead of matching individual line segment or groups of lines based on their topology relationships, we adopt the idea in [35] and [36] to match line pairs which represent building rooftops in 3D scenes. The proposed feature includes two adjacent straight lines and their intersection, which could potentially correspond to a vertex and two connected edges of a building rooftop. Similar to the 3CS features proposed in [36], our feature extraction starts with a line segment detection algorithm. The length of each line segment is then extended according to the edge map, such that the entire boundary of a rooftop can be detected in both images. Intersections of lines are then obtained according to certain line intersecting rules. Lines without intersections or short lengths are unlikely to be robust over two images, and thus are eliminated. The proposed features are then built based on intersections using a permutation method. Although, the number of detected features is much less compared to 2DOC features in [35], they are more geometrically robust and representative of building structures of the scenes.

In the matching stage, the proposed features are not only compared by their geometrical traits, such as lengths and orientations, but also compared using the image context information surrounded by the features. The surrounded regions are distinctiveness enough to identify the true correspondences from the false ones. Meanwhile, the image region that is surrounded by each of the proposed feature, potentially, represents the entire surface of a building rooftop and thus is guaranteed to be planar without any depth discontinuity. Therefore, such regions have a higher chance of being visible in both images which are taken under larger viewpoint variations.
Chapter 3

Methodology

In this chapter, a novel feature called 2EC is introduced. Details of 2EC features and the method for their extraction are first presented. Next, a method for establishing 2EC match correspondences between two oblique aerial images under large projective transformation is presented.

3.1 Introduction

This section briefly introduces the input image dataset, followed by a detailed description of the proposed algorithm.

3.1.1 Input Data

The oblique aerial images from Pictometry International Corp. dataset are used as the input of the proposed algorithm. All the aerial images in our dataset are 2672 × 4008 pixels or 420 × 640 m² with a ground sample distance (GSD) of about 0.15 meter/pixel. These aerial images are captured at an angle from an elevated position. The image acquisition system includes four cameras, with each obliquely downward pointing to north, west, south and east. The input images are captured via an airplane flying over the region of interest in a zigzag way. Considering the way these images are obtained, there may be several images taken from north, west, south and east of each particular scene of interest. Since the images are taken at an average of 50 degrees pitch angle, any two images of the same scene which are captured from different directions undergo large projective variation. An example of an
oblique aerial image pair can be seen in Figure 3.1.

A file called metadata is provided with each image. Information provided in this file includes the flight information, the internal and external parameters of the camera at the time of image capture. A sample metadata file is also provided in Figure 3.2. The camera orientation and location provided in the metadata file have limited accuracy and relying on them solely to establish correspondences between two aerial images is not practical. However, they can be still used to reduce the search space in finding match correspondences. Once the initial correspondences are established with our proposed algorithm, the camera parameters can be refined.

![Figure 3.1: An example of a pair of oblique aerial images. (a) The image was taken from the north direction of the scene. (b) The image was taken from the west direction of the same scene.](image)

<table>
<thead>
<tr>
<th>camera_latitude</th>
<th>49.2011°</th>
</tr>
</thead>
<tbody>
<tr>
<td>camera_longitude</td>
<td>-123.0779°</td>
</tr>
<tr>
<td>camera_height</td>
<td>1.0750e+03m</td>
</tr>
<tr>
<td>camera_pitch</td>
<td>50.2297°</td>
</tr>
<tr>
<td>camera_roll</td>
<td>-1.1451°</td>
</tr>
<tr>
<td>camera_yaw</td>
<td>-88.9865°</td>
</tr>
<tr>
<td>focal_length</td>
<td>84.8186 mm</td>
</tr>
<tr>
<td>senser_rows</td>
<td>2672</td>
</tr>
<tr>
<td>senser_columns</td>
<td>4008</td>
</tr>
<tr>
<td>senser_width</td>
<td>36.072 mm</td>
</tr>
<tr>
<td>senser_height</td>
<td>24.048 mm</td>
</tr>
</tbody>
</table>

Figure 3.2: A sample of metadata file.
3.1.2 Overview of the Proposed Method

Figure 3.3: Flowchart of the proposed system.

Figure 3.3 depicts the flowchart of our proposed algorithm. First, two aerial images from the dataset, along with their metadata files are loaded. Clearly, these images must have some overlaps. We refer to these two images as $I_1$ and $I_2$. 2EC features are then extracted in these two images. In the matching stage, a transformation that is estimated using the metadata (camera parameters in particular) is applied to each feature in $I_1$ to estimate its location in $I_2$. Next, a search space is initiated around the estimated location to find candidate correspondences in $I_2$. The candidate correspondences are then examined and refined via feature shape and image correlation. Once the putative matches are obtained, the outliers are filtered out using a projective matrix optimization method. Based on the
results of the optimization process, a second round of matching is conducted within a more confined search space that leads to a more accurate match establishment.

### 3.2 2EC Feature Extraction

The main idea behind 2EC features is to utilize the geometrical traits of straight edges on the man-made structures to provide distinctive features, which can be robustly matched between input images even under large projective transformations. Here, 2 Edges and one Corner (2EC) feature is introduced, which consists of two connected line segments and their intersection. 2EC features could correspond to the boundaries of building rooftops. In the matching stage, both the appearance and geometrical properties of 2EC features are utilized for a more robust match correspondence establishment.

#### Figure 3.4: Flowchart of 2EC feature extraction.

The flowchart of 2EC features extraction is shown in Figure 3.4. First, Burns [43] line detection and linking algorithm is utilized to detect line segments in each image. The detected lines are then removed or extended according to the edge map. Next, lines are merged, and their intersections and endpoints are found. Three cleaning processes are then performed to remove small, redundant or insignificant lines. Finally, 2EC features are constructed based on the intersection of every two line segments. Detailed procedures are presented in the following sections.

#### 3.2.1 Straight Line Extraction

The objective of this step is to extract a set of straight-line segments from each image. The algorithm implemented to achieve this goal is the Burns line detector [43], which utilizes both the gradient magnitude and gradient orientation to form line support regions and eventually straight line segments. The following steps describe this procedure [44]:

- **Line detection and linking (Sections 3.2.1 and 3.2.2)**
- **Line refinement and extension (Section 3.2.4)**
- **Line merge and defining intersections and endpoints (Section 3.2.4)**
- **Cleaning short, redundant and insignificant lines (Section 3.2.4)**
- **2EC features extraction (Section 3.2.4)**

```plaintext
Input: • Aerial Image

Output: • 2EC features
```

...
1. Partition the pixels into bins based on the gradient orientation values. A bin size of 45 degrees was selected. This results in eight bins being used, and pixels are assigned to bins according to the rules set out in Table 3.1.

<table>
<thead>
<tr>
<th>Bin Number</th>
<th>Gradient Orientation (GO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$0^\circ \leq GO &lt; 45^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>$0^\circ \leq GO &lt; 90^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>$90^\circ \leq GO &lt; 135^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>$135^\circ \leq GO &lt; 180^\circ$</td>
</tr>
<tr>
<td>5</td>
<td>$180^\circ \leq GO &lt; 225^\circ$</td>
</tr>
<tr>
<td>6</td>
<td>$225^\circ \leq GO &lt; 270^\circ$</td>
</tr>
<tr>
<td>7</td>
<td>$270^\circ \leq GO &lt; 315^\circ$</td>
</tr>
<tr>
<td>8</td>
<td>$315^\circ \leq GO &lt; 360^\circ$</td>
</tr>
</tbody>
</table>

2. Run a connected-components algorithm to form line support regions from groups of 4-connected pixels that share the same gradient orientation bin (as shown in Figure 3.5(a)).

3. Eliminate line support regions that have an area smaller than $t_{lsr}$. Given the line support regions shown in Figure 3.5(a) and an area threshold of $t_{lsr} = 3$ pixels, regions 3, 5, and 6 would thus be removed. In the implementation of our algorithm, we want to detect as many lines as possible for the later processing, although many incorrect lines may be included. In the feature extraction procedure, useful lines can be identified, while those lines that incorrectly represent the structure can be rejected with more certainty. In order to keep a balance between the above principle and the computational cost issue, $t_{lsr}$ is set to 10 pixels in the implementation of the algorithm. Please refer to Appendix B for the values of all the parameters mentioned in this chapter.

4. Repeat steps 1, 2, and 3 by shifting the gradient bins to produce a second set of line support regions. This accounts for the possibility that some true lines may have component pixels that lie on either side of an arbitrary gradient orientation boundary (e.g. $45^\circ$ in Table 3.1). Shifted partition bins are shown in Table 3.2. The resulting gradient orientation partitioning and line support regions are shown in Figure 3.5(b).

<table>
<thead>
<tr>
<th>Bin Number</th>
<th>Gradient Orientation (GO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$-22.5^\circ \leq GO &lt; 22.5^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>$22.5^\circ \leq GO &lt; 67.5^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>$67.5^\circ \leq GO &lt; 112.5^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>$112.5^\circ \leq GO &lt; 157.5^\circ$</td>
</tr>
<tr>
<td>5</td>
<td>$157.5^\circ \leq GO &lt; 202.5^\circ$</td>
</tr>
<tr>
<td>6</td>
<td>$202.5^\circ \leq GO &lt; 247.5^\circ$</td>
</tr>
<tr>
<td>7</td>
<td>$247.5^\circ \leq GO &lt; 292.5^\circ$</td>
</tr>
<tr>
<td>8</td>
<td>$292.5^\circ \leq GO &lt; 337.5^\circ$</td>
</tr>
</tbody>
</table>

5. Use the following voting scheme [43] to select preferred lines from the two sets (i.e.
original set and shifted set) of candidate lines:

(a) Line lengths are determined for each region.

(b) Because each pixel is a member of two regions (one in the line support region image and the shifted version), every pixel votes for and is associated with that region of the two that provides the longest interpretation.

(c) Each region is associated with a count of the number of its pixel.

(d) Each region is given a support which is equal to the percentage of the total number of pixels voting for it. The regions selected are those that have a majority support (in this work the support that is greater than 50% ).

6. For each line support region, compute the line represented by that region by performing a least squares fit. The least-squares fit estimates planar model of each line using the
CHAPTER 3. METHODOLOGY

3.2.2 Line Linking

The objective of this step is to link collinear line segments that are separated by very small gaps. Following algorithm describes the linking process [44]:

1. Sort the lines in the order they would be encountered if a horizontal sweep was performed across the image.

2. Use a divide-and-conquer method to efficiently determine nearby pairs of lines.

3. Test each pair of nearby lines to determine whether they should be linked. Conditions (i) and (ii) and one of the conditions of (iii) and (iv), which are illustrated in Figure 3.7, must be satisfied for a pair of lines to be linked. The values of parameters, $d_l$, $t_a$, $t_{overlap}$ and $t_{underlap}$, are set based on the recommendations in [44]. Relaxing the threshold values of $d_l$, $t_a$, $t_{overlap}$ and $t_{underlap}$ can decrease the number of detected lines by grouping lines that indeed are not connected. Besides, increasing the threshold $t_a$ gradient magnitude values. Figure 3.6 represents the fitted lines for the previous example case.
in condition (\(ii\)) could cause line segments to have imprecise locations, due to the grouping of lines that are not co-linear. On the contrary, tightening the thresholds \(d_l\), \(t_{\text{overlap}}\) and \(t_{\text{underlap}}\) can lead to more incomplete lines.

### 3.2.3 Edge Map Creation

Edge map serves as a cue to decide on the extension of the length of a line and its potential intersections. First, the original image is processed by a Canny edge detector [45], followed by dilation of each edge point with a 3 × 3 pixels square structuring element to widen edges and fill out narrow gaps. The Canny edge detector includes three parameters: standard deviation of the Gaussian filter for smoothing and high sensitivity and low sensitivity thresholds. For this stage, they are set to \(\sqrt{2}, 0.1\), and 0.04 respectively. Next, isolated blobs that are smaller than \(t_{\text{iso}}\) pixels are removed. In aerial images, smaller blobs usually correspond to protuberant objects or texture on the surface of rooftops and they could mislead us when extending lines. A value that is equivalent to \(2.25m^2\) in the 3D world is chosen for \(t_{\text{iso}}\), since most of the isolated blobs under this value are redundant on the edge map.

### 3.2.4 Feature Extraction

Once the edge map is obtained, the 2EC features are extracted using the following 6 steps:

1. **Line extension:** The objective of this step is to adjust the detected line segments according to the image edge map such that most of the corresponding edges of rooftops are covered, while as many as nonsignificant lines (lines on the grass regions or grounds) are removed. The extension of a line, on the other hand, should not be too long as it could incorrectly represent edges of multiple buildings along the same line and at a close proximity of each other. Only with meaningful and true line segments, the 2EC features can be constructed robustly. According to this principle, the line extension procedure has the following three steps:

   I) From the set of detected lines, the lines with less than \(t_{\text{pline}} = 80\%\) overlap with the edge map are identified and removed. The remaining lines are kept for future processing steps. A sample of edge map is shown in Figure 3.8, with its detected lines overimposed.
II) To cover for those cases where partial occlusion or low contrast between the rooftop and the background create breakage in the line segments, where they indeed correspond to complete building rooftop edges, each line segment is extended on both ends until it no longer overlaps with the edge map. A maximum extension of \( \min(l, l_{E_{\text{max}}}) \) is allowed, where \( l \) is the length of the line segment. \( l_{E_{\text{max}}} \) is set to a value that is equivalent to 45 meters in the 3D world. This value is chosen to be half of the longest building edge in our image. Also, the image boundaries are used in the cases where the extensions of the line fall outside the image region.

III) Once all lines are extended, the endpoints of each extended line segment is expected to be in close proximity of rooftop vertices if it is from a real rooftop. However, in some cases, small discontinuities exist between the lines’ endpoints and the vertices which correspond to the rooftop corners. Those corners are supposed to be reached by further continuation of the lines. As a result, the extension of each line is continued by \( \min(l'/10, l_{E_{\text{min}}}) \) pixels, where \( l' \) is the length of the extended line from the last step and \( l_{E_{\text{min}}} \) is set to a value that is equivalent to 3 meters in the 3D world.

2. Line merge: In aerial images of urban scene, a rooftop edge often consists of several parallel lines, with each of them representing some parts of the edge. In order to avoid such ambiguity, nearby parallel lines are fused together to form one single continuous line that uniquely corresponds to the full length of the rooftop edge. If all the following three criteria are satisfied, the two line segments with such conditions are merged.
together.

I) They are parallel or almost parallel (a maximum cross angle of $t_{ma} = 5$ degrees is tolerated).

II) The lateral distance between two lines is less than $d_{ml}$ pixels, which is equivalent to 0.45 meter in the 3D world.

III) They have at least 1 pixel overlap.

After merging, the original two line segments are substituted with the new merged line segment. The merging process is applied iteratively until no line segments can be merged together using the above three criteria. Figure 3.9 shows an example of the line merge process. Short lines in the middle of Figure 3.9(a) are merged iteratively into a unique continuous line (green) in Figure 3.9(b), and all those short lines are removed.

![Figure 3.9: Example of line merge.](image)

3. Line intersection: With all the remaining line segments after the extension and merging procedures, the intersections of each line with all the other lines are computed. An intersection is recorded only if it satisfies the following two conditions:

I) The two intersected lines have an angle larger than $\theta_{min} = 20$ degrees and less than $\theta_{max} = 160$ degrees.
II) The intersection point lands on the edge map.

Note that each intersection together with its two associated line segments could potentially correspond to a rooftop vertex and its two edges.

4. Endpoints redefinition: The definition of a 2EC feature includes two rooftop contour line segments and their intersection, which is obtained using previous steps. The contour line segment starts from the intersection and ends at one of the endpoints of each line. However, the endpoints of each line might not be stable or might not accurately occur at its right place (rooftop vertices). Therefore, it is necessary to redefine the endpoints for each line. For this purpose, for each line with at least one intersection, we identify all Harris corners [22] that lie along the line and within 2 pixels lateral distances from it. Three parameters are used in Harris corner detector: standard deviation of smoothing Gaussian, response threshold, and sensitivity factor. In our work, they are set to 1.5, 0.01, and 0.04 respectively. Once all the corners that satisfy the above conditions are identified, they are projected onto their closest lines to form potential locations where lines may end. All the projected points, together with all the intersections on each line are sorted and the most left and most right points are chosen as the two endpoints of each line. The length of each line is then defined by the distance between its two endpoints.

\[ l_1 \quad A \quad B \quad C \quad D \quad l_2 \quad l_3 \]

Figure 3.10: Intersections of \( l_1 \) with \( l_2 \) and \( l_3 \) (B and C respectively). A and C are the projections of the two Harris corner points along \( l_1 \). The endpoints of \( l_1 \) are redefined by A and D.

5. Removal of unstable lines: Here, all the unstable lines are identified and removed. Based on our experiments and observations, we classify three cases as unstable. Lines with at least one of these three conditions are labelled as unstable and are removed iteratively until no more lines can be removed.

I) Lines with only one intersection and no endpoint: Figure 3.11(a) shows line \( l_1 \)
Figure 3.11: a) Lines with no endpoint are removed, b) such removal affects the definition of intersecting lines, c) lines with close intersections are removed, d) short lines creating small chains are removed.

with one intersection and no endpoint. This line along with its intersection with \( l_2 \) are removed (Figure 3.11(b)). The new endpoints of line \( l_2 \) are re-established (\( C \) and \( B \) in the example).

II) Lines with all of their intersections and endpoints very close to each other: As shown in Figure 3.11(c), in the case of \( l_1 \), the length of \( |AB| \) is smaller than \( t_{\text{closeinter}} \) pixels, which is set to a value equivalent to 1.5 meters in the 3D world. Since \( l_1 \) is too short and it is unlikely to correctly represent a rooftop edge, it will be removed, together with all of its intersections. Clearly, the elimination of \( l_1 \) leads to changes in \( l_2, l_3, l_4, \) whose endpoints are updated.

III) Short lines connected together forming a chain: As shown in Figure 3.11(d), all lines, \( l_1 \) to \( l_5 \), are rather short. This chain is unlikely to correspond to the outline of a true rooftop. For each line, whose length is shorter than \( t_{\text{short}} \) pixels, we check to see whether all of its intersected lines are shorter than \( t_{\text{short}} \) pixels. If this is the case, we continue to check whether the second level intersected lines (i.e. the lines that intersect with those lines that intersect with the current queried line) are shorter than \( t_{\text{short}} \) pixels. If all the conditions are satisfied, the current queried line segment is removed. For the example in Figure 3.11(d), we start from \( l_1 \). Since \( l_1 \) and its intersected line \( l_2 \), as well as the second level
interacted line, $l_3$, are all shorter than $t_{\text{short}}$, $l_1$ is removed. $t_{\text{short}}$ is set to a value, which is equivalent to 3 meters in the 3D world.

6. 2EC feature extraction: At each intersection of a line, one or more features can be constructed. As shown in Figure 3.12, $G$ is the intersection of $l_1$ and $l_2$. $G_{13}$, $G_{12}$, $G_{11}$, and $G_{21}$, $G_{22}$ are all the other intersections and endpoints on $l_1$ and $l_2$. For $G$, there are 6 possible combinations of two line segments. In each combination, if both segments are longer than $t_{\text{fex}}$ pixels, the feature will be generated. In Figure 3.12, since $GG_{13}$ is shorter than $t_{\text{fex}}$, it cannot be used to generate features. As a result, only four 2EC features are generated, which share the same intersection point $G$, but with different line segments combinations: $G_{11}GG_{21}$, $G_{11}GG_{22}$, $G_{12}GG_{21}$ and $G_{12}GG_{22}$. In the implementation, $t_{\text{fex}}$ is set to a value, which is equivalent to 4.5 meters in the 3D world.

![Figure 3.12: Feature extraction for the intersection point $G$. Four 2EC features can be extracted.](image)

In a 2EC feature, we refer to the intersection as the center point, and the two endpoints of the lines are referred as end points. As suggested in its name, the center point of 2EC feature could represent the corner of a rooftop where two rooftop boundary lines intersect. Considering the way 2EC features are extracted, more than one 2EC features may share a common center point, while they have line segments with different lengths. Since the center points are the most stable components of 2EC features (due to being rotational and scale invariant), they are used as the point correspondences after 2EC feature correspondences are established.

Figure 3.13 shows examples of 2EC feature extraction on a building. The final results are shown in Figure 3.13(i). In this figure, red circles represent the center points of 2EC features, the yellow line segments are the lines attached to the center points, and the blue stars are the end points of 2EC features. Although there are spurious features on the side
Figure 3.13: a) The original image, b) line detection and linking results, (c) the edge map, d) partially overlapped lines are removed, e) line extension results, f) line merge results, g) lines with their endpoints and intersection redefined, h) unstable lines are removed, i) final 2EC features.
face of the building or the grass region, those features that correspond to the rooftop outline are all extracted. These features can be distinguished from the outliers in the matching stage using their geometrical and image context information.

3.3 Locating Feature Correspondences Between Aerial Image Pairs

Once 2EC features are extracted from the two images, feature correspondences can be established. Two assumptions are made here:

I) Each feature in one image can match at most with one feature in the other image.

II) Not every feature has a correspondence.

For each feature in the first image, a search space is created by predicting the location of that feature in the other image using the acquisition parameters in the metadata files. Both features' shapes and image content are used to locate correspondences within the search space. Once the potential matches are obtained, a projective matrix optimization method is used to reject outliers. The matching process is repeated in an iterative way within smaller search spaces, which is computed based on the result of the first round of the projective matrix optimization.

3.3.1 Background of Multiple View Geometry

In this section, a brief description of the multiple view geometry is reviewed [46]. First, the camera model is presented, followed by the explanation of the epipolar geometry and fundamental matrix. Finally, the computation of projective matrix between two image planes is reviewed.

3.3.1.1 Camera model

Let $X = (X_w, Y_w, Z_w, 1)^T$ be a homogeneous coordinate of a point in the world coordinate system, and $x = (u, v, 1)^T$ the projection of $X$ on an image plane. The mapping from $X$ to $x$ can be expressed as [46]:

$$x = PX$$  \hspace{1cm} (3.1)
In Equation 3.1, $P$ is the $3 \times 4$ homogeneous camera projection matrix, which can be computed from Equation 3.2. $K$ is the calibration matrix, which is computed using the internal camera parameters [46]. In Equation 3.3, $f$ is the focal length. $d_x$ and $d_y$ represent the distance between pixels in the metric world in the $x$ and $y$ directions. $(P_x, P_y)^T$ is the coordinate of the principal point in the image. $R$ and $t$ are the rotation and translation transformations from the world coordinate system to the camera coordinate system. For each point $X$ in the world, its homogeneous coordinate in the camera coordinate system is represented by:

$$X_{cam} = \begin{bmatrix} R^{-1} & -R^{-1}t \\ 0 & 1 \end{bmatrix} X$$

3.3.1.2 Epipolar geometry and fundamental matrix

Given a pair of images of the same scene, a geometric relation must be satisfied between any pair of corresponding points in the two images. This relation is known as epipolar geometry [46]. For each point $x$ in one image, it can be mapped to a line $l'$ in the other image by a $3 \times 3$ matrix $F$. The point $x'$ in the second image corresponding to the point $x$ must also lie on this line $l'$. The same holds true for the reverse. $F$ is known as the fundamental matrix and $l'$ is the epipolar line of point $x$. The following equations are satisfied for any pair of corresponding points $x \leftrightarrow x'$ in the two images:

$$l' = Fx$$
$$1 = F^T x'$$
$$x'^T F x = 0$$

The fundamental matrix is independent of the scene structure, and depends only on the cameras' internal parameters and their relative pose [46]. As explained in [46], given the two camera matrices, $P$ and $P'$, the fundamental matrix can be computed as:
\[ F = [e']_x P' P^+ \]  
where \( P^+ \) is the pseudo-inverse of \( P \), i.e. \( PP^+ = I \). \( e' = (e'_1, e'_2, e'_3)^T \) is the homogeneous coordinate of the epipole in the second image and \([e']_x\) is the skew-symmetric matrix of \( e' \) defined as:

\[
[e']_x = \begin{bmatrix}
0 & -e'_3 & e'_2 \\
-e'_3 & 0 & -e'_1 \\
e'_2 & e'_1 & 0
\end{bmatrix}
\]

### 3.3.1.3 Projective transformation estimation

Given two corresponding point sets \( P = \{p_i, i = 1, \cdots, N\} \) and \( Q = \{q_i, i = 1, \cdots, N\} \). The coordinate of \( p_i \) in \( I_1 \) is represented by \( p_i = [x_i, y_i]^T \) and the corresponding coordinate of \( q_i \) in \( I_2 \) is represented by \( q_i = [u_i, v_i]^T \). The projective transformation \( H \) that maps each \( p_i \) to \( q_i \) can be computed using a least squared algorithm, which is explained in [46]. Consider \( H \) as:

\[
H = \begin{bmatrix}
a & b & c \\
d & e & f \\
g & h & 1
\end{bmatrix}
\]

(3.10)

For each corresponding point pair \( q_i \leftrightarrow p_i \), since \( q_i = H p_i \), two equations can be obtained:

\[
u_i = \frac{a x_i + b y_i + c}{g x_i + h y_i + 1}
\]

(3.11)

\[
v_i = \frac{d x_i + e y_i + f}{g x_i + h y_i + 1}
\]

(3.12)

By multiplying the denominator on both side of the equations, we get:

\[
a x_i + b y_i + c - g u_i x_i - h u_i y_i = u_i
\]

(3.13)

\[
d x_i + c y_i + f - g v_i x_i - h v_i y_i = v_i
\]

(3.14)

Let \( h \) be a column vector:

\[
h = \begin{bmatrix}
a & b & c & d & e & f & g & h
\end{bmatrix}^T
\]

(3.15)

Then Equations (3.13) and (3.14) can be represented in a matrix form:

\[
\begin{bmatrix}
{x_i} & {y_i} & 1 & 0 & 0 & 0 & -x_i u_i & -y_i u_i \\
0 & 0 & {x_i} & {y_i} & 1 & -x_i v_i & -y_i v_i
\end{bmatrix} h = \begin{bmatrix}
u_i \\
v_i
\end{bmatrix}
\]

(3.16)
CHAPTER 3. METHODOLOGY

Each pair of correspondence \( q_i \leftrightarrow p_i \) gives two independent constraints as in Equation (3.16). The following equation can be obtained by putting all the constraints given by all pairs of point correspondences together:

\[
Ah = b
\]

(3.17)

where \( A \) is a \( 2N \times 8 \) matrix, and \( b \) is a \( 2N \times 1 \) column vector. If \( N = 4 \), the matrix \( A \) is invertible, so \( h \) can be computed as:

\[
h = A^{-1}b
\]

(3.18)

However, since usually \( N \) is larger than 4, Equation 3.17 is over-determined. The solution of \( h \) is then computed using a least squared minimization solution [46]:

\[
h = (A^T A)^{-1} A^T b
\]

(3.19)

3.3.2 Overview of the Matching Process

The following steps briefly describe the proposed matching algorithm:

1. For each 2EC feature \( P_{abc} \) in \( I_1 \), the location of its corresponding center point in \( I_2 \) is predicted using the camera parameters in metadata file (Section 3.3.3).

2. A search space is created around the predicted location (Section 3.3.4).

3. All pixels inside the polygon of \( P_{abc} \) are transformed into \( I_2 \). Let us denote the transformed feature by \( P_{a'b'c'} \).

4. An exhaustive search is then conducted among all candidate 2EC features that exist in the search space. The transformed feature \( P_{a'b'c'} \) is compared with each feature in that search space. First, the lengths and orientations of each candidate 2EC feature are compared (Sections 3.3.5). For each candidate feature that satisfies the above condition, a matching score is computed based on the gradient image correlation. The higher the matching score, the more similar the two features. The candidate feature with the highest score (higher than a predefined minimum threshold) is chosen as the putative match for \( P_{abc} \). If there is no match, the matching score is set to \(-\text{Inf}\) (Section 3.3.6). If two or more 2EC features share the same center point in \( I_1 \), the one with the highest matching score is chosen.
5. Projective matrix optimization (Equation (3.19)) is used to remove outliers from the putative matches. Next, a new round of matching process is initiated, while the search space is reduced dynamically based on the optimization result (Section 3.3.7).

3.3.3 Feature Location Prediction

In our algorithm, after the original images are loaded, metadata associated with each image are also loaded. The metadata provides additional information, including camera orientation (Yaw, Pitch and Roll angles), camera location (latitude, longitude, and altitude), camera internal parameters, etc. Although these parameters are imprecise, they can still be used to predict the shape and the position of each 2EC feature from one image to the other. In the remaining part of this section, our prediction method is first presented, followed by the evaluation of the prediction accuracy.

3.3.3.1 Feature location estimation using metadata

One basic task in the matching process is to predict a 2EC feature location from $I_1$ to $I_2$. After that, a neighborhood will be created around the predicted location within which the search can be performed. Specifically, consider a pair of images taken by two cameras ($C_1$ and $C_2$) from different viewpoints. $m$ is a point in $I_1$, and $m'$ is its actual corresponding location in $I_2$. $m$ can be transformed from $I_1$ to $I_2$ with a two steps transformation, which is computed using the imprecise camera parameters in the metadata files. The transformed location can then be used as a rough estimation of the location of $m'$.

Two assumptions are made in here. First, since the altitude of the camera (for example, 1058.4 meters is a typical value for the camera's altitude) is much higher than the height of each building, we assume that the building rooftops are located at the surface of the earth. Therefore, the depth of the image can be estimated as the altitude of the camera, and the 3D location of any point in the image can be determined with a single image and its camera parameters. Second, since the ground region covered by two cameras is small compared with the surface of the earth, the ground is considered flat.

Based on these two assumptions, the model for two cameras and the ground region covered by their images are illustrated in Figure 3.14. Two steps are taken for this transformation: first, $m$ in $I_1$ is projected onto the 3D world, next, the 3D point $M$ is projected back into 2D on $I_2$. In order to implement such transformation, a world coordinate is used
as the reference coordinate. Here, the center of the world coordinate system is located at the center of camera 1 ($C_1$). The $X$ axis is pointing to North, the $Y$ axis is pointing to West, and the $Z$ axis is pointing to the Earth Center. This means that the world coordinate system is the same as the first camera coordinate system, when its yaw, pitch, and roll angles are all zero. The transformation of a point $m(x_0, y_0)$ from $I_1$ to the 3D world includes two procedures:

1. Transform the image coordinates of $m(x_0, y_0)$ into the world coordinate system using the internal and external parameters of camera 1:

$$
\begin{bmatrix}
  x_w \\
  y_w \\
  z_w
\end{bmatrix} = R(yaw_1, pitch_1, roll_1) \begin{bmatrix}
  (x_0 - p_{x_1}) \times d_{x_1} \\
  (y_0 - p_{y_1}) \times d_{y_1} \\
  f_1
\end{bmatrix}
$$

The values of $yaw_1$, $pitch_1$, and $roll_1$ are the orientation of camera 1. $(p_{x_1}, p_{y_1})^T$ is the coordinate of the principal point in $I_1$. $d_{x_1}$ and $d_{y_1}$ represent the conversion between the image pixel and metric world. $f_1$ is the focal length of the first camera. These parameters are all provided by the metadata of $I_1$. 3D rotation $R(\alpha, \beta, \gamma)$ is defined by:

$$
R(\alpha, \beta, \gamma) = \begin{bmatrix}
  \cos(\alpha) & -\sin(\alpha) & 0 \\
  \sin(\alpha) & \cos(\alpha) & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  \cos(\beta) & 0 & \sin(\beta) \\
  0 & 1 & 0 \\
  -\sin(\beta) & 0 & \cos(\beta)
\end{bmatrix} \begin{bmatrix}
  1 & 0 & 0 \\
  0 & \cos(\gamma) & -\sin(\gamma) \\
  0 & \sin(\gamma) & \cos(\gamma)
\end{bmatrix}
$$
2. Project \((x_w, y_w, z_w)^T\) onto the ground by connecting it to the camera center and continuing it until it intersects with the ground. The relationship between the projected 3D point \(M(X_0, Y_0, Z_0)\) and \((x_w, y_w, z_w)^T\) can be shown as:

\[
\begin{align*}
\frac{x_w}{X_0} &= \frac{y_w}{Y_0} = \frac{z_w}{Z_0} \\
Z_0 &= \text{alt}_1
\end{align*}
\] (3.22) (3.23)

where \(\text{alt}_1\) is the altitude of camera 1 provided in the metadata of \(I_1\).

Next, according to [46], \(M(X_0, Y_0, Z_0)^T\) can be projected onto \(I_2\) using the camera matrix \(P_2\), which can be computed using the camera parameters in the metadata of \(I_2\):

\[
m' = P_2M, \quad P_2 = K_2\begin{bmatrix} R_2^{-1} & -R_2^{-1}t_2 \end{bmatrix}
\] (3.24)

As explained in [46], in the above equations, \(K_2\) is the calibration matrix computed using the internal camera parameters:

\[
K_2 = \begin{bmatrix} f_2/dx_2 & 0 & px_2 \\
0 & f_2/dy_2 & py_2 \\
0 & 0 & 1 \end{bmatrix}
\] (3.25)

\(R_2\) is the orientation of the camera 2 with respect to the world coordinate system which centered at \(C_1\). \(R_2\) can be computed by:

\[
R_2 = R^{-1}(\text{lon}_1, -\text{lat}_1 - 90, 0)R(\text{lon}_2, -\text{lat}_2 - 90, 0)R(\text{yaw}_2, \text{pitch}_2, \text{roll}_2)
\] (3.26)

where \(\text{lat}_1, \text{lon}_1, \text{lat}_2\) and \(\text{lon}_2\) are the latitude and longitude of the two cameras. \(t_2\) is the coordinate of the second camera, \(C_2\), in the world coordinate system. It can be computed by:

\[
t_2 = R^{-1}(\text{lon}_1, -\text{lat}_1 - 90, 0)(C_{\text{cam}_2} - C_{\text{cam}_1})
\] (3.27)

where \(C_{\text{cam}_1}\) and \(C_{\text{cam}_2}\) represent the 3D Cartesian coordinates of the two cameras with respect to ECEF (Earth-Centered, Earth-Fixed) coordinate system, and they are computed using the longitude and latitude of the two cameras according to [47].

The above algorithm can be applied on each one of the 2EC features in \(I_1\) to predict its location and shape in \(I_2\). In fact, since we consider the ground as a plane, the corresponding points in the two image planes are related by a projective matrix \(H_m\) [46]. In order to find this matrix \(H_m\), the following procedures are performed:
1. Four points are selected in $I_1$: $p_1 = (1, 1)^T$, $p_2 = (1, N)^T$, $p_3 = (M, 1)^T$, $p_4 = (M, N)^T$, where $N \times M$ is the size of $I_1$.

2. Transform these four points to $I_2$ using the method explained above. Let us assume the transformed points as $p'_1$, $p'_2$, $p'_3$, and $p'_4$.

3. With four corresponding points $p_i \leftrightarrow p'_i, i = 1, \ldots, 4$, the projective matrix $H_m$ can be computed using the method explained in Section 3.3.1.3.

During the matching process, the projective transformation $H_m$ is applied to all 2EC features in $I_1$ to predict their locations and shapes in $I_2$.

![Figure 3.15: The transformation of a 2EC feature from $I_1$ to $I_2$.](image)

Figure 3.15 shows an example of the prediction. The feature highlighted in the left image is transformed to the right image using the above transformation. The transformed feature is shown in the right image with red color and its corresponding true location is highlighted with green. One can notice that the predicted location is very close to its true location. Moreover, the lengths and orientations of the two are also similar.

### 3.3.3.2 Evaluation of prediction accuracy

In order to investigate the accuracy of prediction and provide a basis for selecting some of the thresholds required in the matching process, we applied the above transformation on 10 pairs of images. Among them, 5 pairs have a difference of 90 degrees in yaw angle variations, and the other 5 pairs have a difference of 180 degrees. For each pair of images, 120 matched points, which were uniformly spread over the entire image, were selected manually. The points in $I_1$ were transformed to $I_2$ using the method explained in the above section and
Figure 3.16: The histogram of the distances between the predicted locations and the ground truth locations for image pairs with 90 (a) and 180 degrees (b) viewpoint variations.

Figure 3.17: The histogram of the distances from the ground truth locations to their corresponding estimated epipolar lines for image pairs with 90 (a) and 180 degrees (b) viewpoint variations.

Figure 3.18: The histogram of the differences between the length of the transformed line segments and the corresponding ground truth for image pairs with 90 (a) and 180 degrees (b) viewpoint variations.
four measurements were calculated: First, the distances between the transformed points and their ground truth locations were calculated. Second, for each point in $I_1$, its epipolar line was computed according to [46]. The distances from the ground truth locations in $I_2$ to their corresponding epipolar lines were calculated. Third, the lengths of the line segments, which were generated by any combination of two transformed points were computed and compared with that of the corresponding ground truth features. The length difference was measured as a percentage of the ground truth length. Finally, the cross angles between the transformed line segments and their ground truth were computed.

The statistical values for the above measures are provided in Figures 3.16 to 3.19. One can see that the predicted positions have a maximum offset of less than 300 pixels from the ground truth locations. Also, the predicted epipolar lines are less than 150 pixels away from their ground truth locations. Moreover, around 96.58% of the predicted line segments have less than 10% length difference compared with their corresponding ground truth line segments, and around 95.26% of the predicted line segments have less than 5 degrees orientation difference compared with their ground truth line segments. These errors are mainly induced by two reasons. One reason is the imprecise camera parameters provided in the metadata file. Another reason lies in the assumption that all buildings in the images have zero height. According to the epipolar geometry, which was reviewed in Section 3.3.1.2, each point in $I_1$ can be mapped to an epipolar line in $I_2$. The location of the mapped point on the epipolar line, however, depends on the height of this point from the ground. Since
the actual height of points varies in both images, their predicted locations in $I_2$ would have different offsets from their ground truth locations.

### 3.3.4 Search Space Creation

Since the aerial images have a large size ($2672 \times 4008$ pixels), it is very time consuming to search for correspondences over the entire image. Also, searching in a large domain means a higher chance of wrongly identifying matches. Therefore, creating a smaller search space that contains the correct correspondence is very critical to the process. Let $P_{abc}$ represents a 2EC feature in $I_1$. Its center point is denoted by $c_P$ and its two end points are denoted by $a_P$ and $b_P$. The following two steps are designed to create its corresponding search space in $I_2$:

1. Transform the center point $c_P$ to $I_2$ using the method described in Section 3.3.3.
   Let's denote the transformed point by $c'_P$. Create a square neighborhood of size $(2s_{\text{square}} + 1) \times (2s_{\text{square}} + 1)$ pixels, centered at $c'_P$. In our experiments, $s_{\text{square}}$ is selected to be 500 pixels. This value is chosen based on the test results of the prediction accuracy in Section 3.3.3.2.

2. Compute the epipolar line of $c_P$ using the estimated fundamental matrix (Section 3.3.1.2).
   A window along the epipolar line is then created. The width of the window is chosen to be $2s_{\text{window}}$ pixels, where $s_{\text{window}} = 150$ pixels based on the test results of the prediction accuracy in Section 3.3.3.2.

![Figure 3.20: Search space generation.](image)

As shown in Figure 3.20, the intersection of the two areas generated above, denoted by $\Omega_{P_{abc}}$, is utilized as the search space for $P_{abc}$ (the gray area). All the features in $I_2$ with their center points in $\Omega_{P_{abc}}$ will be considered as potential matching candidates for $P_{abc}$. 
CHAPTER 3. METHODOLOGY

3.3.5 Filter Candidate Features Based on Feature Shape

Once the search space $\Omega_{P_{abc}}$ for the feature $P_{abc}$ has been determined, an exhaustive search among all the features inside $\Omega_{P_{abc}}$ will be initiated. The exhaustive search includes two rounds of process in a hierarchy way. First, the candidates are filtered based on the shape of the features. Next, all the qualified candidates will go through a second round of comparison using the image information inside the polygonal shape area of each feature. The first round of search includes the following tasks:

1. Transform $P_{abc}$ to $P_{a'b'c'}$ in $I_2$ using the method explained in Section 3.3.3. The transformed center point is denoted by $c'_P$ and two end points are denoted by $a'_P$ and $b'_P$.

2. For the transformed feature $P_{a'b'c'}$, and each of the candidate features in $\Omega_{P_{abc}}$, a four dimensional descriptor $(l_1, l_2, \alpha_1, \alpha_2)$ is defined. Take $P_{a'b'c'}$ as an example, $l_1 = |a'_Pc'_P|$, $l_2 = |b'_Pc'_P|$, $\alpha_1$ and $\alpha_2$ are the orientations of $|a'_Pc'_P|$ and $|b'_Pc'_P|$ with respect to the $x$ axis of the image (as shown in Figure 3.20).

3. Given two 2EC features with their descriptors of $(l_1, l_2, \alpha_1, \alpha_2)$ and $(l'_1, l'_2, \alpha'_1, \alpha'_2)$, their similarity is determined using the following four equations:

$$\frac{|l_1 - l'_1|}{l_1} < T_{len} \quad (3.28)$$
$$\frac{|l_2 - l'_2|}{l_2} < T_{len} \quad (3.29)$$
$$|\alpha_1 - \alpha'_1| < T_{angle} \quad (3.30)$$
$$|\alpha_2 - \alpha'_2| < T_{angle} \quad (3.31)$$

The thresholds $T_{len}$ is set to 10% (0.1), and $T_{angle}$ is set to 5 degrees in our experiments. The similarity between $P_{a'b'c'}$ and all its candidate features in $\Omega_{P_{abc}}$ are evaluated. Candidates that do not satisfy the above four criteria are rejected while those that satisfy are kept for the succeeding round of comparison.

3.3.6 Filter Candidate Features Based on Image Correlation

After the first round of filtering, for a typical feature in $I_1$, there may be one or more qualified candidate features in $I_2$. In order to identify the feature with the highest similarity among
the remaining candidates, a correlation based method is incorporated. This method is similar to the template matching in which we compute the similarity between corresponding pixels in image patches. However, since the shapes of buildings in the scene are variant and we do not have any prior knowledge of that, using a template-based matching to extract building rooftops and locate correspondences with a method similar to [48] is not an option in this work.

Since for each 2EC feature, composed of three critical points (one center point and two end points), an image patch can be generated by adding a fourth point to form a parallelogram with the other three points. The similarity between 2EC features can be evaluated by computing the correlation of their corresponding image patches. There are two reasons that make the use of correlation possible. 1) The images are taken by cameras from relatively high altitude, so rooftops can be seen in both images, even though they go into projection and deformation. Since 2EC features represent the rooftop profiles of the buildings, they can be extracted in both images. 2) Since the rough camera parameters are provided via the metadata file, the image patches created by 2EC features can be transformed from $I_1$ to $I_2$.

Instead of using pixel intensity in the input images to compute correlation, we use the gradient images. The advantage of using gradient images is that they are more stable than the intensity images under illumination variation. The Sobel operators are used to approximate the first derivative of the image in both horizontal and vertical directions. Magnitude of the gradients are then obtained by summing $|I_x|$ and $|I_y|$ together. The gradient of the input image can be calculated by:

\[
fs = \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1
\end{bmatrix}
\]  
(3.32)

\[
I_x = fs \otimes I
\]  
(3.33)

\[
I_y = fs^T \otimes I
\]  
(3.34)

\[
\nabla I = |I_x| + |I_y|
\]  
(3.35)

The detailed explanation for computing the similarity between $P_{abc}$ in $I_1$ and $Q_{abc}$ in $I_2$ using correlation are given as:
1. Consider that \( \{x_k\} (1 \leq k \leq K) \) are the pixels inside the polygon formed by \( P_{abc} \), where \( K \) is the number of pixels. Transform the coordinate of each pixel inside the polygon to \( I_2 \) using the method detailed in Section 3.3.3. Let us assume the transformed pixels locations are \( \{x'_k\} \) in \( I_2 \), where \( 1 \leq k \leq K \).

2. Compute vector \( \mathbf{v}_t = \mathbf{c}_Q - \mathbf{c}'_P \), where \( \mathbf{v}_t \) is the translation that moves \( \mathbf{c}'_P \) to \( \mathbf{c}_Q \).

3. Translate each \( x'_k \) with the vector \( \mathbf{v}_t \) such that \( \mathbf{c}'_P \) coincides with \( \mathbf{c}_Q \): \( x'_k = x'_k + \mathbf{v}_t \).

4. A similarity value for \( P_{abc} \) and \( Q_{abc} \) is set by computing the correlation of the two sets of image gradient values: \( \nabla I_1(x_k) \) and \( \nabla I_2(x'_k) \), where \( 1 \leq k \leq K \). Specifically, the equations used to calculate the similarity value are:

\[
\begin{align*}
\nabla I_P &= \frac{1}{K} \sum_{k=1}^{K} \nabla I_1(x_k) \quad (3.36) \\
\nabla I_Q &= \frac{1}{K} \sum_{k=1}^{K} \nabla I_2(x'_k) \quad (3.37)
\end{align*}
\]

\[
D_{PQ} = \frac{\sum_{k=1}^{K} [\nabla I_1(x_k) - \nabla I_P][\nabla I_2(x'_k) - \nabla I_Q]}{\sqrt{\sum_{k=1}^{K} [\nabla I_1(x_k) - \nabla I_P]^2} \sqrt{\sum_{k=1}^{K} [\nabla I_2(x'_k) - \nabla I_Q]^2}} \quad (3.38)
\]

A similarity measure is assigned to each of the remaining candidate features in \( \Omega_{P_{abc}} \) using the above process. If the maximum similarity value of all the candidates is larger than a threshold \( t_{corr} \), the corresponding feature is extracted as a match to \( P_{abc} \). Otherwise, there is no match found for \( P_{abc} \). Considering that the prediction procedure in Section 3.3.3 is not a precise one, due to the imprecise camera parameters in the metadata files, \( t_{corr} \) is set to 0.5. Increasing this value would result in higher accuracy of the matching result, with smaller number of match correspondences. An example of the image patch correlation is given in Figure 3.21. Figures 3.21(a) and (b) include two 2EC features (green and red), which must be matched. Figure 3.21(c) is the gradient image of (a), along with the polygonal shape generated by the 2EC feature in (a). Figure 3.21(d) is the transformed version of the green feature in (a). The center of this feature is translated to coincide with the red feature in (b). The similarity score is then computed using the gradient values of the corresponding
pixels in two polygons in (c) and (d). The correlation score in this case is computed to be 0.5666, which indicates a match.

For those 2EC features in $I_1$ that do not have a match in $I_2$, a matching score of -Inf is assigned. If two or more 2EC features in $I_1$ share the same center point, the one with the highest matching score is kept, while the others are ignored. Finally, the center points of the 2EC feature correspondences are selected as the initial set of point correspondences.

![Figure 3.21: 2EC feature matching based on correlation: a) and b) include two image patches, which have 90 degrees viewpoint differences. 2EC features (green and red) in these two images are corresponding features. c) The gradient image of a) along with the polygonal shape generated by the 2EC feature in (a). d) The transformed version of the polygon in (c). The center of this polygon coincides with the center of the 2EC feature in (b).](image)

### 3.3.7 Removal of the Outliers

After the initial set of point correspondences is obtained, a statistical method is used to identify outliers among these point correspondences. Figure 3.22 shows the flowchart of this procedure.
Consider the initial point matches as \( P = \{ p_i, i = 1, \cdots, N \} \) and \( Q = \{ q_i, i = 1, \cdots, N \} \). The coordinate of \( p_i \) in \( I_1 \) is represented by \( p_i = (x_i, y_i)^T \) and the corresponding coordinate of \( q_i \) in \( I_2 \) is represented by \( q_i = (u_i, v_i)^T \). If all the corresponding points in the 3D world are coplanar, the perspective points in two image planes are related by a projective transformation \( H \). In such case, for each corresponding point pairs \( p_i \leftrightarrow q_i \), \( q_i = Hp_i \). For aerial images, the two image plane is not strictly related by a projective transformation because of the height variations of the buildings and terrain. However, since the height variations are very small compared to the altitudes of the two cameras, the two input aerial images can still be related by a homography with certain residual error. The projective matrix \( H \) is computed using the initial matches with the method explained in 3.3.1.3.

With the estimated homography transformation \( H \), points in \( I_1 \) are mapped to \( I_2 \) by \( p_i' = Hp_i \). Let us assume the transformed point \( p_i' = (x_i', y_i')^T \). The distances between the transformed points and their matched locations in \( I_2 \) are calculated as:

\[
s_i = \text{dist}(q_i, p_i') = \sqrt{(u_i - x_i')^2 + (v_i - y_i')^2}
\]

Next, point pairs with their distances larger than \( \bar{s} + t_{fac}\sigma(s) \) are removed as outliers, where \( \bar{s} \) and \( \sigma(s) \) are the mean and standard deviation of the distance set \( \{s_i\} \). The above procedure is repeated iteratively for four times with \( t_{fac} \) equals 3, 2.5, 2 and 1.5 in each iteration. This is performed because of a simple reason: in the first iteration, since the correspondences set may contain more incorrect matches, the homography computed using...
this set contains larger error. Therefore, only those matches with really large residual error (in our case, larger than $\bar{s} + 3\sigma(s)$) are sure to be incorrect and are eliminated. As more iterations are performed, more incorrect matches are rejected. Since the projective matrix becomes more accurate, the criteria for rejecting outliers become more strict, and more matches can be thrown out. Once the above four iterations are exhausted, a new round of iteration is initiated. At each iteration, 5% of the match pairs with the largest distance are removed iteratively and the process stops if one of the following conditions is satisfied: 1) The maximum distance in $\{s_i\}$ is less than $s_{tre} = 30$ pixels. 2) The number of matches is less than $n = 20$.

Once the iteration stops, a new projective transformation $H'$ is computed using the remaining matches. The residual error set $\{s_i\}$ is also updated and the current largest distance in $\{s_i\}$ is found and denoted by $s_{larg}$. Next, the matching process is repeated again, but within a small search space, which is created using $s_{larg}$. Specifically, for each 2EC feature $P_{abc}$ in $I_1$, instead of transforming the center point $c_p$ to $I_2$ using the method explained in Section 3.3.3, we transform it to $I_2$ using $H'$: $c'_p = H'c_p$. The search space is created as a $(2s_{larg} + 1) \times (2s_{larg} + 1)$ square shape centered at the projected point $c'_p$. Since $s_{larg}$ is much smaller than $s_{square}$, the search space is much more confined than the one created in Section 3.3.4. For each feature $P_{abc}$, the searching for its correspondence is then performed within this confined search space.

Once this round of search is done, the final point correspondences are selected as the center points of the 2EC feature correspondences. Figure 3.23 shows a pair of sample images with all the matches identified.

![Figure 3.23: A pair of sample images with all the matches identified.](image)
Chapter 4

Experimental Results

In this chapter, the stability and distinctiveness of 2EC features are investigated. SIFT features detection method is used as a comparison to demonstrate the advantages of the proposed features. Next, the quality of 2EC features is evaluated by registration of slant aerial image that include wide variations in the viewing angles. The superiority of these features is shown by comparing the registration results using state of the art image features. Finally, the implementation related issues along with the discussion of time complexity are presented.

4.1 2EC Features Evaluation

The purpose of this section is assessing the stability and distinctiveness of 2EC features. For this purpose, the 2EC feature detection algorithm was applied on six images, which were cut from larger slant aerial images in our database. The reason that we only focus on small images in this section is to emphasize the stability of 2EC features. 2EC features will be applied on larger aerial images in the next section in order to test their effectiveness in establishing match correspondences. Since the aerial images are taken from different viewpoints, they undergo large yaw and pitch variations. The side faces of the buildings can be seen in some areas, while they are invisible in others. Meanwhile, in some of other regions, the edges of the rooftops are occluded by trees. Also, the images are taken at different times of the year and under different illumination conditions. In order to demonstrate the robustness of the 2EC feature detection, the last two images are down-sampled by a factor of 2 using the first two images. Later in Section 4.2, 2EC features are extracted in
CHAPTER 4. EXPERIMENTAL RESULTS

down-sampled aerial images for a fair comparison with the state of the art, which could not perform correctly with the original images because of their large sizes. From now on, we will refer the test images as image a) ~ image f).

In the remainder part of Section 4.1, the feature detection results are first presented, followed by the matching results which are obtained using 2EC features.

4.1.1 Feature Detection Results

Figure 4.1 shows the extracted 2EC features on the test input images. In the figure, the red circles represent the locations of the center points of 2EC. The yellow line segments are the lines attached to the center points. The blue stars at the end of the yellow lines are the end points of the 2EC features. In all of the input images, the center points of most 2EC features are located on the corners of the building rooftops, with their two associated lines profiling the building rooftop edges. Few features belong to the background regions including cars, vegetations, and sidewalks.

In Figures 4.1(a) and (b), 67 and 61 features were detected respectively. Visually, if the center points of two 2EC features have the same physical locations within 2 pixels from the corresponding true locations, and if their connected lines profile the same edges, they are considered as a repetitive feature pair. The features in both images were checked manually, and 14 repetitive feature pairs were found, all of which were on the scene buildings. In the case of Figures 4.1(c) and (d), one of these two images is much darker (d) than the other (c). Regardless, the feature detection quality was not affected and 13 pairs of repetitive features were found between the two images. As for images in Figures 4.1(e) and (f), since they were down-sampled, the feature detection results were affected by the poorer resolution. There were more erroneous features on tree branches and parked cars. Also, the top-left part of the building in Figure 4.1(e) was not detected as it was in Figure 4.1(a). However, there were still 13 correct repetitive features which were extracted on the buildings in both images.

As a comparison, Figure 4.2 shows the location of SIFT features on the input images. As can be noted, more SIFT features were detected in these images, however, most of the features were located on tree branches and cars. Compared to 2EC features, there were also more SIFT features detected on the buildings. Not only did these features occur on the rooftop corners, but also they appeared on the white protuberant objects on the rooftops. Nonetheless, the quality of the features was severely affected by downsampling. By checking carefully, only five repetitive features were found that belonged to the building rooftops.
Figure 4.1: 2EC features for buildings from different viewpoints: a) captured from south, yaw=−95.6°, pitch=50.4°, b) captured from north, yaw=−88.8°, pitch=50.1°, c) captured from east, yaw=177.5°, pitch=50.7°, d) captured from west, yaw=−0.20°, pitch=50.8°, e) down-sampled version of a), f) down-sampled version of b).
Figure 4.2: SIFT features for buildings from different viewpoints. The tested images are the same as those in Figure 4.1.
Despite the higher number of SIFT features detected in our input images, they are not distinctive enough and therefore establishing correct match correspondences between image pairs is very challenging if not impossible. We will investigate the distinctiveness of 2EC features and compare it with that of SIFT features by matching the input images in the next section.

4.1.2 Feature Matching Results

To demonstrate the uniqueness of 2EC features, we matched Figures 4.1(a) and (b) using the extracted 2EC features in the last subsection. The number of match correspondences and the matching accuracy were computed for each case. The results were then compared with the matching performance using SIFT features on the same pair of images. The matching procedures were slightly different for 2EC and SIFT features. We will describe them in the following section respectively.

4.1.2.1 2EC features matching

For our proposed feature, we first extracted 2EC features separately in the input images. The results can be seen in Figures 4.1(a) and (b). As explained in the previous chapter, the description of 2EC feature includes two parts: the first part includes the length and orientation of a feature, and the second part includes the image region surrounded by the four sided polygonal shape. After we transformed each feature from image a) to image b) using a rough estimate of the transformation (based on the metadata of each input image), we compared the lengths and the directions of the two lines in the transformed feature with every feature in image b). Next, for those features in image b), which had similar shapes with the transformed feature, we compared the similarity of the image regions surrounded by the four sided polygons. Specifically, the gradient image correlation was computed for every qualified 2EC feature in image b). The feature with the maximum correlation score which was larger than a threshold $t_{corr}$ was then selected as a putative match. In this experiment, $t_{corr}$ was set to 0.1. Finally, the outliers were removed using the projective matrix optimization method. The details of the above procedures were described in Section 3.3. As shown in Figure 4.3, fourteen correspondences were found between two images, and all of them were matched correctly.
4.1.2.2 SIFT features matching

For SIFT features [18], the descriptor is a 128 dimensional vector that is computed from the image gradient magnitude and orientation over the $64 \times 64$ pixels region around the keypoint location. We had detected SIFT features and extracted descriptors from the input images, and matched them with the ratio distance method suggested in [18]. However, since the two images underwent severe projective transformation, there was no match found between the two. Considering that in our proposed method, we took advantage of the imprecise camera parameters provided with the input images, it was only fair to use this information for matching SIFT features. Therefore, we applied the following procedures:

First, to compensate for the deformation caused by the viewpoint variation, the projective and affine transformations were removed from the input images using the camera parameters with the method suggested in [1]. The detail of this method will be briefly explained later in Section 4.2.2.2. As shown in Figure 4.4, this step made the two input images to have the same directions, and the lengths and orientations of all the corresponding straight lines to become almost similar. Next, SIFT features and descriptors were extracted in the transformed images. Finally, match correspondences were established by comparing the Euclidean distances in the descriptor space. The result is shown in Figure 4.5. Despite of the high repetitiveness of SIFT features, as shown in Figures 4.4(a) and (b), there were only two match correspondences found between the two images, and among them one was incorrect. Since the number of matches was not enough for computing a projective matrix (a minimum of eight matches is required), we could not process this image any further as we had for our 2EC features.
CHAPTER 4. EXPERIMENTAL RESULTS

Figure 4.4: Two images after the removal of the projective and affine transformations along with the detected SIFT features on them: a) Correspondence to Figure 4.2(a), b) correspondence to Figure 4.2(b). The red marks highlight the SIFT features.

Figure 4.5: Results of SIFT matching for two images. 1 out of 2 matches is incorrect.

4.2 2EC Feature Application On Aerial Images Matching

In this section, we will evaluate the quality of 2EC features through establishing match correspondences in oblique aerial images. Considering the way these images are acquired, for one subject scene, there would be several images taken from North, East, South and West directions. If image pairs are captured from the similar directions, they will have similar yaw angles and therefore they mostly undergo a linear translation and a small projective transformation. For these images, it is easy to process and establish match correspondences and usually most existing methods work well. However, two oblique aerial images taken from
two different directions, for example, North and East, result in large projective transformation, which is very challenging for all the state of the art matching algorithms. Therefore, we only focus on image pairs with severe yaw angle variations from now on. Fifteen pairs of images were chosen from our database. Among them, five image pairs have a difference of 90 degrees in yaw angle, five pairs have a difference of 180 degrees, and the other five pairs have 270 degrees difference. All these images have pitch angles over 40 degrees. The combination of yaw and pitch angles variations leads to large projective transformations between these images.

For each image pair, 2EC features were first extracted (Section 3.2) and initial correspondences were established (Sections 3.3.2~3.3.6). Then the incorrect matches (outliers) were removed using optimized projective matrix. Finally, the matching procedures were repeated one more time based on a new search space that was created in each case using the optimized projection matrix (Section 3.3.7). The original size of each image is $2672 \times 4008$ pixels. In our experiments, in order to reduce the running time and for the convenience of comparison with the state of the art, we have down-sampled the input images by a factor of 2. All the parameters that were used in the experiments were kept the same for all the input images. The parameters and their settings are presented in Appendix B.

### 4.2.1 Experimental Results

The results generated for the fifteen pairs of images are summarized in Table 4.1. In this table, the entry Image Pair represents the number of the image pairs and the difference between the two cameras in yaw angles. The column Features provides the number of 2EC features detected in each image (separated by a hyphen) with the form $I_1-I_2$. The GT column represents the number of Ground Truth correspondences that are manually detected in both images. The next three entries of IC, CC, CR respectively present the number of Initial Correspondences, the number of Correct Correspondences, and the percentage of Correct matches or the Ratio of CC/IC. The statistics of the final results after the projective matrix optimization are presented in the next succeeding three entries (PMC, PCC, PCR). The last column of the table, Err, presents the accuracy of the established matches in pixels computed using the epipolar geometry. In other words, this column represents the average perpendicular distance in pixels from the ground truth points to their computed epipolar lines. The measurement is estimated using the following steps:
Table 4.1: Results of match correspondence establishment in oblique aerial images using 2EC features.

<table>
<thead>
<tr>
<th>No.</th>
<th>Image Pair</th>
<th>Feature (#)</th>
<th>GT (#)</th>
<th>IC (#)</th>
<th>CC (#)</th>
<th>CR (%)</th>
<th>PMC (#)</th>
<th>PCC (#)</th>
<th>PCR (%)</th>
<th>Err (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90_1</td>
<td>1879 – 1415</td>
<td>140</td>
<td>100</td>
<td>92</td>
<td>92.00</td>
<td>90</td>
<td>88</td>
<td>97.78</td>
<td>1.3</td>
</tr>
<tr>
<td>2</td>
<td>90_2</td>
<td>2431 – 2047</td>
<td>58</td>
<td>67</td>
<td>40</td>
<td>59.70</td>
<td>36</td>
<td>34</td>
<td>94.44</td>
<td>6.2</td>
</tr>
<tr>
<td>3</td>
<td>90_3</td>
<td>2512 – 1772</td>
<td>75</td>
<td>67</td>
<td>56</td>
<td>83.58</td>
<td>44</td>
<td>43</td>
<td>97.73</td>
<td>1.2</td>
</tr>
<tr>
<td>4</td>
<td>90_4</td>
<td>1186 – 1292</td>
<td>45</td>
<td>33</td>
<td>31</td>
<td>93.94</td>
<td>32</td>
<td>31</td>
<td>96.88</td>
<td>4.6</td>
</tr>
<tr>
<td>5</td>
<td>90_5</td>
<td>2095 – 3084</td>
<td>57</td>
<td>57</td>
<td>47</td>
<td>82.46</td>
<td>41</td>
<td>39</td>
<td>95.12</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>180_1</td>
<td>1393 – 1333</td>
<td>95</td>
<td>48</td>
<td>44</td>
<td>91.67</td>
<td>48</td>
<td>48</td>
<td>100.00</td>
<td>1.7</td>
</tr>
<tr>
<td>7</td>
<td>180_2</td>
<td>1843 – 2503</td>
<td>60</td>
<td>43</td>
<td>33</td>
<td>76.74</td>
<td>35</td>
<td>32</td>
<td>91.43</td>
<td>2.4</td>
</tr>
<tr>
<td>8</td>
<td>180_3</td>
<td>2560 – 3241</td>
<td>94</td>
<td>69</td>
<td>58</td>
<td>84.06</td>
<td>55</td>
<td>54</td>
<td>98.18</td>
<td>1.9</td>
</tr>
<tr>
<td>9</td>
<td>180_4</td>
<td>1874 – 1823</td>
<td>59</td>
<td>43</td>
<td>34</td>
<td>79.07</td>
<td>37</td>
<td>35</td>
<td>94.59</td>
<td>2.1</td>
</tr>
<tr>
<td>10</td>
<td>180_5</td>
<td>738 – 1046</td>
<td>51</td>
<td>35</td>
<td>32</td>
<td>91.43</td>
<td>33</td>
<td>32</td>
<td>96.97</td>
<td>2.3</td>
</tr>
<tr>
<td>11</td>
<td>270_1</td>
<td>2238 – 1871</td>
<td>106</td>
<td>89</td>
<td>75</td>
<td>84.27</td>
<td>82</td>
<td>79</td>
<td>96.34</td>
<td>2.5</td>
</tr>
<tr>
<td>12</td>
<td>270_2</td>
<td>3334 – 2084</td>
<td>81</td>
<td>45</td>
<td>41</td>
<td>91.11</td>
<td>40</td>
<td>39</td>
<td>97.50</td>
<td>1.5</td>
</tr>
<tr>
<td>13</td>
<td>270_3</td>
<td>887 – 932</td>
<td>58</td>
<td>41</td>
<td>37</td>
<td>90.24</td>
<td>32</td>
<td>31</td>
<td>96.88</td>
<td>2.8</td>
</tr>
<tr>
<td>14</td>
<td>270_4</td>
<td>1470 – 1750</td>
<td>92</td>
<td>70</td>
<td>60</td>
<td>85.71</td>
<td>70</td>
<td>67</td>
<td>95.71</td>
<td>1.5</td>
</tr>
<tr>
<td>15</td>
<td>270_5</td>
<td>738 – 1046</td>
<td>85</td>
<td>76</td>
<td>64</td>
<td>84.21</td>
<td>60</td>
<td>58</td>
<td>96.67</td>
<td>1.3</td>
</tr>
</tbody>
</table>

1. Compute the fundamental matrix $F$ over the input image pairs using the normalized 8-point algorithm [46] and the initial point correspondences obtained by matching the 2EC features.

2. For each ground truth pair $cp_{1i} \leftrightarrow cp_{2i}, i = 1, \ldots, N$ the epipolar lines of $cp_{1i}$ and $cp_{2i}$ are computed using the fundamental matrix $F$. The equations used to compute the epipolar lines are shown below:

$$l_i = F \cdot cp_{1i} \quad (4.1)$$

$$l'_i = F^T \cdot cp_{2i} \quad (4.2)$$

3. The distance of $cp_{2i}$ from its corresponding epipolar line $l_i$ and the distance of $cp_{1i}$ from its corresponding epipolar line $l'_i$ are measured. The average residual error is computed by:

$$Err = \frac{1}{2N} \sum_{i=1}^{N} \left( d(cp_{2i}, l_i) + d(cp_{1i}, l'_i) \right) \quad (4.3)$$

where $d(cp, l)$ is the distance of the point $cp$ from the line $l$, $N$ is the number of ground truth points. $Err$ measures the average distance from the ground truth points to their computed...
epipolar lines.

As shown in columns $PMC$, $PCC$, and $PCR$ of Table 4.1, even though the number of true correspondences is relatively small, the correct ratio in all image pairs is higher than 90%. An average correct rate of 96.41% is achieved, with an average residual error of 2.32 pixels. Since the purpose of establishing correspondences in aerial images is often to reconstruct 3D models, 8 pairs of match correspondences with high accuracy is sufficient for computing fundamental matrix. Therefore, as long as the number of correspondences is larger than eight, the results are considered meaningful. However, higher number of correct matches generally leads to more accurate fundamental matrix. Besides, if the match correspondences are uniformly distributed all over the image, the fundamental matrix computed using them will be more accurate. Since the ground truth correspondences used for the above measurements are spread all over the entire image, the residual error in the last column of the table can reflect the accuracy of the fundamental matrix computed using the found matches. It basically reflects the accuracy and reliability of those found match correspondences.

Before the projective matrix optimization, the correct ratio is relatively low as shown in the entry $CR$. The projective matrix optimization step removes most of the outliers and the successive iteration of the matching adds more true matches to the final results. Compared to the percentage shown in the column of $CR$, the results in $PCR$ have been improved significantly. It should be noted that some of the outliers could not be filtered out as they are too close to the locations of true matches.

Figure 4.6 shows an example of oblique aerial image pair with the correspondences established using 2EC features. The two images have a difference of about 180 degrees in yaw angles and each has a pitch angle of about 50 degrees. The epipolar geometry is verified in Figure 4.6(b). The three red dots are randomly selected. They represent the points that are used to generate the epipolar lines in both images. It can be seen that the corresponding epipolar lines pan through the same image locations. The fifteen pairs of input images and their epipolar lines generated using the found match correspondences are displayed in Figures A.1 $\sim$ A.15 in Appendix A.

### 4.2.2 Comparison with the State of the Art

In order to compare the quality of 2EC features with that of the state of the art, three popular features including SIFT, SURF, and ASIFT were used. We evaluated the quality of the features by finding match correspondences between images of wide view variations.
Figure 4.6: Results of matching using 2EC features for an aerial image pair with 180 degrees view differences: a) The correspondences established by the proposed algorithm. There are 48 matches, and all of them are found correctly. b) The epipolar lines generated by the red dots.

For this purpose, the same fifteen image pairs were chosen as described in Section 4.2.

4.2.2.1 Summaries of the comparison algorithms

In this section, the SIFT, SURF, and ASIFT feature extraction and description algorithms are summarised respectively.

1. SIFT [18], features which were first introduced by Lowe, are claimed to be invariant with respect to image scaling and rotation. However, it was noticed that as the view angle variation between the input images increases, the quality of the matching decreases. The correct matching rate is only about 50% when there is a 50 degrees viewpoint variation between the images [18]. To extract the SIFT features, scale space
images are first generated by repeatedly convolving the Gaussian kernels of variable deviations with the input image. Next, Difference of Gaussian (DoG) images are produced by subtracting the adjacent scale space images. The keypoints are then localized by comparing each pixel with its neighbors in a $3 \times 3$ pixels region at the current and adjacent levels of DoG images. In order to extract the descriptor for a keypoint, the gradient magnitude and orientation of each image pixels in a $16 \times 16$ pixels region around the keypoint is computed. In each $4 \times 4$ pixels sub-region, the gradient orientation is voted into 8 bins and the gradient magnitudes are then accumulated according to their orientations. Based on a voting scheme, a SIFT descriptor is then formed by a 128 dimensional vector. For establishing the correspondences, Lowe suggested to use the distance ratio method [18]. The match correspondences are obtained by comparing the Euclidean distance of the closest neighbor to that of the second closest neighbor.

2. SURF [19] is first proposed by Bay et al. in 2006. SURF is claimed to be more robust and several times faster than SIFT. The keypoint detection of SURF is based on the Hessian matrix. Specifically, the local maxima of the determinant of the Hessian matrix are selected as the keypoints. Given an image $I$, the Hessian matrix at point $x$ is defined as

$$H(x) = \begin{bmatrix} L_{xx}(x) & L_{xy}(x) \\ L_{xy}(x) & L_{yy}(x) \end{bmatrix},$$ (4.4)

where $L_{xx}(x), L_{xy}(x), L_{yy}(x)$ are the convolution of the Gaussian second order derivatives in $xx, yy,$ and $xy$ directions with the image $I$. In [19], box filters are used to provide an approximation for the Gaussian second order derivatives. An example of the box filters is shown in Figure 4.7. The initial image $I$ is convolved with the box filters of increasing size to obtain the scale space images. The determinant of Hessian matrix is then computed using images at each scale levels. To localize the keypoints, the local maxima of the determinants are searched in a $3 \times 3$ pixels region at the current and adjacent scale levels. To extract the descriptor of a keypoint, a dominant orientation is found first. The Haar wavelet response is computed in a circular region centered at each keypoint. The dominant orientation is then determined by summing all the response inside a slide window of size 45 degrees. Next, a square region is selected around the keypoint oriented along the dominant orientation. This region is divided into four $4 \times 4$ pixels sub-regions. In each sub-region, Haar wavelet response
is computed and a 4 dimensional vector is extracted, which represents the wavelet response in the x and y directions. By combining the vectors in all the sub-regions, a 64 dimensional vector is formed as the SURF descriptor. During the matching stage, in order to increase the processing speed, descriptors are compared with each other in the Euclidean space only if the two features have the same sign of Laplacian (the trace of the Hessian matrix).

![Figure 4.7](image)

Figure 4.7: The first row (from left to right): the second order derivatives of the Gaussian filter in the $xx$, $yy$ and $xy$ directions. The standard deviation of the Gaussian filter is 1.2, and the size of the filter is $9 \times 9$ pixels. The second row (from left to right): the box filters corresponding to the second order derivatives of Gaussian filter in the $xx$, $yy$ and $xy$ directions. The value of the gray area in the box filters equals to 0, and the values of the black and white areas are shown in each figure. Figure source: [19].

3. ASIFT [5] is claimed to be fully affine invariant and significantly outperform SIFT. [5] shows that ASIFT works up to 80 degrees latitude angle rotations. However, no quantitative matching accuracy is reported in [5]. Specifically, ASIFT generate a set of sample images which simulate all possible affine transformations of the input images by varying the orientation of the camera optical axis. These affine transformations depend on two parameters: the latitude $\theta$ and the longitude $\phi$. $\theta$ angle is the angle between the camera optical axis and the norm to the image plane and $\phi$ is the angle between the plane which is made by the optical axis and the norm and a fix plane which is vertical to the image plane. To simulate an affine transformation, the input
image is first rotated by \( \phi \) followed by a sub-sampling of \( t \) pixels in the \( x \) direction of the image, where \( t \) is defined as \( t = \left| \frac{1}{\cos(\theta)} \right| \). All simulated images for the two input images are then compared with each other using SIFT. The pair of the images with the largest number of match correspondences is selected and the corresponding matches are transformed back to the initial image. The Optimized Random Sample Algorithm (ORSA) [25] is used to eliminate inconsistent matches according to the epipolar geometry constraint.

### 4.2.2.2 Application of comparison features on aerial images matching

To make a fair comparison between the proposed 2EC features and the above mentioned features, we utilized the acquisition parameters in the metadata file (that include imprecise camera parameters) accompanying the input images in the matching process. Specifically, the acquisition parameters were used to rectify the projective and affine transformations in both images. Also, the search space was selected based on the acquisition parameters to reduce the running time and increase the accuracy of the matching process. In the remaining part of this section, we will describe the matching procedures in details.

For all the three features of SIFT, SURF and ASIFT, projective and affine transformations were first removed from the input images using the method suggested in [46] and [1]. Generally, projective transformation recovery makes the parallel lines in the real world parallel in the image. Affine transformation removal recovers the real world right angles in the image. In summary, eight steps were applied to each input image to remove the projective and affine transformations. It should be noted that these procedures were adapted from [1].

1. Four points in the real world \( c_1 = (1,1,0)^T \), \( c_2 = (2,1,0)^T \), \( c_3 = (1,2,0)^T \), \( c_4 = (2,2,0)^T \) were projected to the image plane using the camera matrix \( P \), which was computed using the camera parameters provided in the metadata file (Section 3.3.1.1). The projected points were denoted by \( c'_1 \), \( c'_2 \), \( c'_3 \), \( c'_4 \).

2. Two sets of lines were then computed using the four projected corners: \( e_1 = c'_1 \times c'_2 \), \( e_2 = c'_3 \times c'_4 \), \( m_1 = c'_2 \times c'_3 \), \( m_2 = c'_1 \times c'_4 \). The lines \( e_1 \) and \( e_2 \), \( m_1 \) and \( m_2 \) were parallel in the real world, but intersected at points \( p_1 \) and \( p_2 \) in the image, where \( p_1 = e_1 \times e_2 \) and \( p_2 = m_1 \times m_2 \).

3. The vanishing line of the image was computed by connecting \( p_1 \) and \( p_2 \): \( L = (l_1, l_2, l_3)^T = \ldots \)
4. The matrix $H_p$, which was applied to the input image to remove the projective transformation, was then computed as:

$$H_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix}$$

(4.5)

5. Since $e_1$ and $m_1$, $e_2$ and $m_2$ were two sets of orthogonal lines in the real world, $s_{11}$ and $s_{12}$ could be computed from the following equations:

$$\begin{bmatrix} e_1^1m_1^1 & e_1^1m_1^2 + e_1^2m_1^1 \\ e_2^1m_2^1 & e_2^1m_2^2 + e_2^2m_2^1 \end{bmatrix} \begin{bmatrix} s_{11} \\ s_{12} \end{bmatrix} = \begin{bmatrix} -e_1^2m_1^2 \\ -e_2^2m_2^2 \end{bmatrix}$$

(4.6)

6. A symmetric matrix $S$ was then computed, from which the SVD of $S$ was obtained:

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{12} & 1 \end{bmatrix}$$

(4.7)

$$S = UDV^T$$

(4.8)

7. The affine transformation $H_a$ that was used to restore the affine transformation could be computed as:

$$H_a = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}$$

(4.9)

$$A = U\sqrt{D}V^T$$

(4.10)

8. Apply $H_p$ and $H_a$ to $I_1$ and $I_2$: $I'_1 = H_{a1}H_{p1}I_1$ and $I'_2 = H_{a2}H_{p2}I_2$.

A rotation $R$ was then applied to $I'_2$ using the yaw angle difference between the two input images so that $I'_1$ and $I'_2$ would have the similar directions. In summary, the entire transformations from $I_1$ to $I'_1$ and $I_2$ to $I'_2$ could be denoted by $T_1 = H_{a1}H_{p1}$ and $T_2 = RH_{a2}H_{p2}$. After the above transformations, the lengths and orientations of all the corresponding straight lines in both images became almost similar [1].

Features and descriptors were then extracted in the transformed image pairs $I'_1$ and $I'_2$. The detection and description methods for each algorithm was as mentioned in Section 4.2.2.1. For each feature in $I'_1$, its match correspondence was then searched within a
search space in $I_2$. The search space was generated using the camera parameters in the metadata files accompanying the input images. The detailed description of creating the search space is presented below:

1. Considering a feature $X_1$ located at $x'_1$ in the transformed image $I'_1$, its corresponding location in the original image $I_1$ was computed as $x_1 = T^{-1}_1 \begin{bmatrix} x'_1 & 1 \end{bmatrix}^T$.

2. Created the search space in $I_2$ for the point $x_1$ in $I_1$ using a rough estimation of transformation based on the camera parameters in the metadata files as described in Sections 3.3.3 and 3.3.4. This step is the same as what was performed for search space creation for 2EC features. For now, let the four vertices of the search space in $I_2$ be denoted by $a_1, a_2, a_3, a_4$.

3. Mapped the four corners to $I'_2$: $a'_1 = T_2 \begin{bmatrix} a_1 & 1 \end{bmatrix}^T$, $a'_2 = T_2 \begin{bmatrix} a_2 & 1 \end{bmatrix}^T$, $a'_3 = T_2 \begin{bmatrix} a_3 & 1 \end{bmatrix}^T$, $a'_4 = T_2 \begin{bmatrix} a_4 & 1 \end{bmatrix}^T$. The area inside the polygon which was created by $a'_1, a'_2, a'_3, a'_4$ was considered as the search space for feature $X_1$.

For the algorithms of SIFT and SURF, the above procedures were applied to each feature detected in $I'_1$ to create a search space in $I_2$. The match correspondence was then searched inside the search space. The image pair in Figure 3.1 after removing the projective and
affine transformations and view angle rotation are shown in Figure 4.8. A SIFT feature is displayed in $I'_1$, and its corresponding search space is shown in $I'_2$. As for ASIFT, the search for the match correspondences was performed over the entire image.

4.2.2.3 Experimental results

Table 4.2 to 4.4 show the matching results for each of the above three algorithms. To assess the quality of the results, the accuracy of matching was inspected manually. The number of correspondences, the number of correct correspondences, and their ratio are given in the columns of Total Matches, Correct Matches, and Correct Rate. The average residual errors are provided in the last column of Error.

As shown in Table 4.2, SIFT achieved an average correct matching rate of 86.32%. Comparing to Table 4.1, one can see that 2EC features have higher correct matching rate than SIFT features in all cases. In order to perform a fair comparison between 2EC and SIFT features, the matching accuracy of the SIFT feature was further verified using the method explained in Section 4.2.1. Specifically, the fundamental matrix was calculated using the match correspondences. The residual error was computed by averaging the distances of the ground truth points from their corresponding epipolar lines. Smaller error in here implies that the epipolar geometry established using the match correspondences is more accurate. For SIFT features, the average residual error over 15 pairs of input images is 10.81 pixels. Comparing the last column of Table 4.1 with that of Table 4.2, one can observe that the error for 2EC features is less than that of SIFT features in all the cases except Row 3, which has very similar values for both 2EC and SIFT features. In conclusion, the proposed 2EC feature outperforms SIFT by an average correct rate of 10.09% in all the cases except one.

As shown in Table 4.3, the average correct matching rate for SURF is 75.40%, with an average residual error of 22.85 pixels. In all the cases of SURF, the correct matching rate and the residual error are worse than the proposed 2EC feature. As for ASIFT feature, it has an average correct matching rate of 85.14%, with an average residual error of 68.07 pixels. It, however, outperforms 2EC feature in two cases (Row 3 and 13) with less residual error and similar correct matching rates. As shown in Figures A.3 and A.13, due to abundant texture on the ground and some rooftops of these two pairs of images, feature correspondences were easily identified using ASIFT features. Beside, since its match correspondences spread all over the images, ASIFT resulted in a fundamental matrix with higher accuracy and less residual error.
CHAPTER 4. EXPERIMENTAL RESULTS

Table 4.2: Results of match correspondences establishment using SIFT features.

<table>
<thead>
<tr>
<th>No.</th>
<th>Image Pair</th>
<th>Total Matches (#)</th>
<th>Correct Matches (#)</th>
<th>Correct Rate (%)</th>
<th>Err (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90_1</td>
<td>95</td>
<td>87</td>
<td>91.58</td>
<td>2.7</td>
</tr>
<tr>
<td>2</td>
<td>90_2</td>
<td>40</td>
<td>30</td>
<td>75.00</td>
<td>36.7</td>
</tr>
<tr>
<td>3</td>
<td>90_3</td>
<td>257</td>
<td>250</td>
<td>97.28</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>90_4</td>
<td>66</td>
<td>59</td>
<td>89.40</td>
<td>9.2</td>
</tr>
<tr>
<td>5</td>
<td>90_5</td>
<td>160</td>
<td>152</td>
<td>95.00</td>
<td>3.5</td>
</tr>
<tr>
<td>6</td>
<td>180_1</td>
<td>153</td>
<td>149</td>
<td>97.39</td>
<td>2.6</td>
</tr>
<tr>
<td>7</td>
<td>180_2</td>
<td>81</td>
<td>73</td>
<td>90.12</td>
<td>2.5</td>
</tr>
<tr>
<td>8</td>
<td>180_3</td>
<td>20</td>
<td>6</td>
<td>30.00</td>
<td>13.2</td>
</tr>
<tr>
<td>9</td>
<td>180_4</td>
<td>245</td>
<td>223</td>
<td>91.02</td>
<td>3.0</td>
</tr>
<tr>
<td>10</td>
<td>180_5</td>
<td>50</td>
<td>36</td>
<td>72.00</td>
<td>18.2</td>
</tr>
<tr>
<td>11</td>
<td>270_1</td>
<td>43</td>
<td>37</td>
<td>86.05</td>
<td>51.4</td>
</tr>
<tr>
<td>12</td>
<td>270_2</td>
<td>20</td>
<td>19</td>
<td>95.00</td>
<td>4.7</td>
</tr>
<tr>
<td>13</td>
<td>270_3</td>
<td>209</td>
<td>199</td>
<td>95.22</td>
<td>6.0</td>
</tr>
<tr>
<td>14</td>
<td>270_4</td>
<td>106</td>
<td>100</td>
<td>94.34</td>
<td>6.4</td>
</tr>
<tr>
<td>15</td>
<td>270_5</td>
<td>22</td>
<td>21</td>
<td>95.45</td>
<td>1.6</td>
</tr>
</tbody>
</table>

Table 4.3: Results of match correspondences establishment using SURF features.

<table>
<thead>
<tr>
<th>No.</th>
<th>Image Pair</th>
<th>Total Matches (#)</th>
<th>Correct Matches (#)</th>
<th>Correct Rate (%)</th>
<th>Err (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90_1</td>
<td>51</td>
<td>44</td>
<td>86.27</td>
<td>18.6</td>
</tr>
<tr>
<td>2</td>
<td>90_2</td>
<td>85</td>
<td>75</td>
<td>88.24</td>
<td>64.0</td>
</tr>
<tr>
<td>3</td>
<td>90_3</td>
<td>200</td>
<td>171</td>
<td>85.50</td>
<td>4.8</td>
</tr>
<tr>
<td>4</td>
<td>90_4</td>
<td>86</td>
<td>56</td>
<td>65.12</td>
<td>14.8</td>
</tr>
<tr>
<td>5</td>
<td>90_5</td>
<td>138</td>
<td>131</td>
<td>94.93</td>
<td>2.0</td>
</tr>
<tr>
<td>6</td>
<td>180_1</td>
<td>70</td>
<td>65</td>
<td>92.86</td>
<td>11.0</td>
</tr>
<tr>
<td>7</td>
<td>180_2</td>
<td>57</td>
<td>51</td>
<td>89.47</td>
<td>4.9</td>
</tr>
<tr>
<td>8</td>
<td>180_3</td>
<td>12</td>
<td>8</td>
<td>66.67</td>
<td>26.9</td>
</tr>
<tr>
<td>9</td>
<td>180_4</td>
<td>173</td>
<td>159</td>
<td>91.91</td>
<td>2.9</td>
</tr>
<tr>
<td>10</td>
<td>180_5</td>
<td>37</td>
<td>30</td>
<td>81.08</td>
<td>44.6</td>
</tr>
<tr>
<td>11</td>
<td>270_1</td>
<td>20</td>
<td>6</td>
<td>30.00</td>
<td>45.4</td>
</tr>
<tr>
<td>12</td>
<td>270_2</td>
<td>15</td>
<td>6</td>
<td>40.00</td>
<td>47.1</td>
</tr>
<tr>
<td>13</td>
<td>270_3</td>
<td>226</td>
<td>196</td>
<td>86.73</td>
<td>10.9</td>
</tr>
<tr>
<td>14</td>
<td>270_4</td>
<td>47</td>
<td>41</td>
<td>87.23</td>
<td>6.6</td>
</tr>
<tr>
<td>15</td>
<td>270_5</td>
<td>20</td>
<td>9</td>
<td>45.00</td>
<td>38.2</td>
</tr>
</tbody>
</table>
Table 4.4: Results of match correspondences establishment using ASIFT features.

<table>
<thead>
<tr>
<th>No.</th>
<th>Image Pair</th>
<th>Total Matches (#)</th>
<th>Correct Matches (#)</th>
<th>Correct Rate (%)</th>
<th>Err (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90_1</td>
<td>403</td>
<td>390</td>
<td>96.77</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>90_2</td>
<td>491</td>
<td>403</td>
<td>82.08</td>
<td>13.3</td>
</tr>
<tr>
<td>3</td>
<td>90_3</td>
<td>1366</td>
<td>1331</td>
<td>97.44</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>90_4</td>
<td>251</td>
<td>147</td>
<td>58.57</td>
<td>29.4</td>
</tr>
<tr>
<td>5</td>
<td>90_5</td>
<td>979</td>
<td>919</td>
<td>93.87</td>
<td>1.6</td>
</tr>
<tr>
<td>6</td>
<td>180_1</td>
<td>560</td>
<td>480</td>
<td>85.71</td>
<td>19.2</td>
</tr>
<tr>
<td>7</td>
<td>180_2</td>
<td>213</td>
<td>190</td>
<td>89.20</td>
<td>4.1</td>
</tr>
<tr>
<td>8</td>
<td>180_3</td>
<td>0</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>9</td>
<td>180_4</td>
<td>1023</td>
<td>924</td>
<td>90.32</td>
<td>2.5</td>
</tr>
<tr>
<td>10</td>
<td>180_5</td>
<td>149</td>
<td>125</td>
<td>83.89</td>
<td>9.8</td>
</tr>
<tr>
<td>11</td>
<td>270_1</td>
<td>53</td>
<td>0</td>
<td>0</td>
<td>n/a</td>
</tr>
<tr>
<td>12</td>
<td>270_2</td>
<td>32</td>
<td>18</td>
<td>56.25</td>
<td>730.0</td>
</tr>
<tr>
<td>13</td>
<td>270_3</td>
<td>999</td>
<td>964</td>
<td>96.50</td>
<td>0.9</td>
</tr>
<tr>
<td>14</td>
<td>270_4</td>
<td>214</td>
<td>195</td>
<td>91.12</td>
<td>3.5</td>
</tr>
<tr>
<td>15</td>
<td>270_5</td>
<td>52</td>
<td>0</td>
<td>0</td>
<td>n/a</td>
</tr>
</tbody>
</table>

4.2.3 Discussion of the Experimental Results

In this section, the advantages of 2EC features are discussed based on the experimental results, which were presented in Sections 4.2.1 and 4.2.2. The successful matching between the fifteen pairs of images with large projective transformations demonstrates that a large viewpoint variation between two images can be overcome using 2EC features. Because of the viewpoint variation, the side faces of the buildings can be seen in one view; however they disappear in the other views. Nonetheless, 2EC features can be extracted on rooftops which are common in both views.

Also, the center points of 2EC features can be located at the corners of building rooftops with high accuracy, which leads to the high accuracy of matching process. Moreover, match correspondences established using 2EC features can spread all over the images as long as there are man-made structures in the common regions between the two scenes. This makes the fundamental matrix computed using the matching results more accurate. In contrast, the essence of SIFT, SURF, and ASIFT features is to locate keypoints, around which large intensity variations occur. When applied to the aerial images, these features always appear on the stains on the ground or the protuberant objects on the rooftops or sides. Such locations are harder to be identified precisely in both images. Moreover, match correspondences
established using SIFT, SURF and ASIFT features only concentrate in small areas of abundant intensity variations, but they are absent in flat areas where surfaces are more uniform and textureless. Both these two disadvantages of SIFT, SURF and ASIFT features can induce error to the fundamental matrix calculation process. Figure 4.9 shows an example of matching using 2EC and SIFT features. Although both resulted in high correct matching rate (100% for 2EC feature and 97.39% for SIFT features), the residual error for SIFT was higher than that of 2EC (2.6 pixels for SIFT features and 1.7 pixels for 2EC features).

Compared to SIFT, SURF, and ASIFT features, 2EC features are more distinctive. SIFT, SURF, ASIFT only describe features using local regions around them. It is rather hard for such descriptors to be distinctive enough when there are similar patterns in an image. However, 2EC features incorporate physical properties of lines as well as image
data, and therefore can be more uniquely identified. Take image pairs 90.2, 270.1 and 270.5 (Figures A.2, A.11 and A.15) for example. Similar patterns are very common in these images. For all these cases, 2EC features outperformed SIFT, SURF, and ASIFT with higher correct matching rate and less error.

Finally, the robustness of 2EC features to the illumination variation is demonstrated by the cases of image pairs 180.3 and 270.2. As can be seen in Figure A.8, there is a large illumination change between the image pairs of 180.3. SIFT, SURF had very low correct matching rate and ASIFT failed to find any match at all. Nonetheless, 55 2EC features were matched successfully, out of which 54 were indeed correct. In the image pair 270.2 (Figure A.12), the two images are both dark and have low contract. Regardless, 40 matches were successfully identified using 2EC features, with a matching accuracy of 97.5% and an average residual error of only 1.5 pixels. In contrast, SIFT had a higher residual error and SURF and ASIFT had much lower correct matching rates.

4.2.4 Limitation of 2EC Features

Straight lines are very common in man-made structures. They dominate the industrial areas, airports, campus, and high-rises. In such scenes, 2EC features can be easily detected and they represent the geometrical traits of man-made structures well. However, as for the scenes which are dominated by smaller residential houses (especially those with gabled rooftops), 2EC features are more difficult to identify. There are several reasons for this problem. First, 2EC features are hard to extract on the rooftops of the houses, due to the occlusion by trees or the irregular shape of the rooftops. Also, since the rooftops are gabled, one surface can be seen in one view, but it will invisible or undergo large change visually in the other view. As a result, it is more difficult to have common 2EC features that describe the same structure. Moreover, in suburban areas, many of the houses in each neighbourhood include same design traits, making unique identifications of 2EC features very hard if not impossible. Clearly, in such cases, 2EC features are not distinctive enough. However, not only is this type of images difficult for 2EC features, but also for other existing features.

The image quality has a great influence on the 2EC feature extraction. If the image is too dark or has too low resolution, the 2EC features extraction will be affected and so will the matching process. A better image quality will lead to more 2EC features and more match correspondences. In our experiments, we resized original images by a factor of 0.5 in each dimension. This means that the ground sample distance (GSD) is reduced from
0.15m/pixel to 0.3m/pixel. This is done for a more fair comparison with the state of the art, which could not perform correctly when dealing with the original images due to their large size. A higher correct matching rate of around 98% with a less residual error of 2 pixels could be achieved if the original images are used for 2EC features extraction and matching process.

As described in Section 3.3.3, the feature location prediction is based on the assumption that the ground plane is flat, and the height variations of the buildings on the ground is very small compared to the altitude of the two cameras. These two assumptions partially could explain the prediction error, which was tested in Section 3.3.3. However, when there are high-rise buildings in the scene, the second assumption is violated. In such cases, the predicted location of a 2EC feature on top of the high-rise building may have larger error and may even fall out of its search space. To solve this issue, an even larger search space should be created. Besides, in the final stage of matching process, the outliers are rejected using a projective matrix model, which is also adopted based on the above two assumptions. This model is rather simple, and can also be violated when there are high-rise buildings in the scene.

The proposed feature extraction and matching algorithm employ some parameters and thresholds that need to be set according to the characteristics of the input images. The parameter setting for our input dataset and portability issues are presented in Appendix B. In order to adapt the algorithm for other type of images, those parameters need to be adjusted accordingly.

### 4.3 Discussion of the Time Complexity

All the code for extracting 2EC features and matching process are implemented in MATLAB (R2010a). The platform which is used to test the program is a PC with an Intel Core 2 Duo (3GHz) processor, 8.00 GB of RAM, and 64-bit windows operating system. Since 2EC features extraction is based on the line detection algorithms and the match procedure is an exhaustive search, the entire program is time consuming. The executive time highly depends on the size of the input images. The average time for establishing match correspondences for a pair of images with 1336 × 2004 pixels is 1314.3 seconds (22 minutes). To make the running time much faster, the program can be implemented in C++. The execution time for each stage of the algorithm was measured for each of the fifteen image pairs and the
average time is presented in the remaining part of this section.

First, an average of 9.5 seconds is taken to load the input image pairs, together with their accompanying metadata file. This procedure also includes the Harris corners detection, the gradient images computation, the imprecise fundamental matrix computation using the camera parameters in the metadata file, and the other relative parameters configuration. Next, an average of 438.5 seconds (7.3 minutes) is spent on the line detection and line linking procedures. 2EC features for both images are then extracted in 295.7 seconds (5 minutes) in total.

At the first round of matching stage, a large search space is chosen for each 2EC feature such that high matching rate can be achieved. Since several thousands of times comparing between feature candidates are required, the process is rather slow. The average time for the first round of matching is 507.4 seconds (8.5 minutes). Next, an average of 0.01 seconds is spent on the projective matrix optimization. Finally, based on the result of the optimization, the second round of matching is conducted within a smaller search space, which leads to a much faster process. The average running time is 63.2 seconds (1 minute).
Chapter 5

Conclusions

In this work, 2EC features were proposed for the purpose of establishing match correspon-
dences between oblique aerial images, which undergo large projective transformations. A
complete solution for matching 2EC features was also proposed. The established correspon-
dences could be used to accurately compute the fundamental matrix and obtain refined
camera parameters that are used for image registration or 3D reconstruction purposes.

The extraction of 2EC features required detecting straight lines in the input images and
these lines were utilized as a medium to extract viewpoint invariant corners. Both lines and
corners were encapsulated in the definition of 2EC features, such that each feature could
potentially correspond to a vertex and two connected edges of a building rooftop. The
geometrical characteristics of 2EC features ensured their surrounded image regions to be
planar, and therefore viewpoint invariant under large viewpoint variations.

In the matching process, both geometrical and visual properties of each feature were used
to establish match correspondences in two images of same scene taken from substantially
different views. In order to ensure that the matching process is robust to illumination
changes, image gradient values in the viewpoint invariant regions were used to compute the
similarity scores for 2EC features. After putative matches were obtained, a technique based
on projective matrix optimization was introduced to efficiently remove outliers and refine
the transformation parameters.

The experimental results showed the effectiveness of 2EC features to locate match cor-
respondences between oblique aerial images under large projective transformation. The
superiority of 2EC features was demonstrated by comparing it with three other state of the
art features, including SIFT, SURF and ASIFT. Based on the experimental results, two
conclusions were made:

1. The feature extraction method proposed in Chapter 3 enables 2EC features accurately profile the boundaries of rooftops in aerial images. 2EC features extraction is robust with respect to viewpoint and illumination variations. The experimental results showed that the match correspondences established by 2EC features can result in more accurate fundamental matrix compared to the correspondences established through other features such as SIFT, SURF and ASIFT.

2. The special structure of 2EC features enables it to encapsulate both the geometrical and local image information to uniquely describe scene features. The regions surrounded by 2EC features are invariant to viewpoint variations, since they mostly correspond to flat rooftops. Due to their large sizes, they are distinctive enough to be identified accurately.

The following section summarizes the important contributions made in this thesis and points out possible future research directions.

5.1 Summary of Contributions

In this thesis, two contributions were made:

1. We proposed 2EC features that hold geometrical traits of straight edges and their intersections with each other and are especially designed to represent typical configurations in man-made structures. Since 2EC features preserve geometrical relationships between object edges, they can be associated with each other even under large projective transformations that may make them look different in different views.

2. A new matching algorithm was developed that utilizes both visual and geometrical properties in a hierarchical way to robustly and accurately establish match correspondences between two or more images of a scene.

5.2 Potential Future Research

Possible future research directions of the proposed 2EC feature are listed below:
1. Both the feature extraction and the matching algorithms are rather slow, due to the large size of input images and the nature of the matching algorithm, where the exhaustive search is required among large number of candidate features. The running time can be significantly reduced by implementing the entire program using C++ instead of MATLAB.

2. The 2EC features work well for buildings with flat rooftops. However, for buildings with gabled rooftops or for smaller residential houses, the extraction of 2EC features is problematic. Therefore, the proposed algorithm is limited to scenes that are dominated by structures with flat rooftop surfaces. In order to detect 2EC features on small residential houses correctly, images with higher resolution can be used. Besides, more suitable features should be developed that are robust with respect to smaller size residential buildings with gabled rooftops.

3. Lack of distinctiveness could be another problem that 2EC features suffer from when it comes to matching smaller residential houses. A graph based feature grouping and matching methodology could be developed in order to remove the matching ambiguities, caused by the similar houses in the suburban regions. Such method should be able to cluster nearby features based on their locations, feature characteristics and the topological relationship between neighboring features.

4. The application of 2EC features can be extended to the general wide baseline matching problems in man-made scenes. For example, walls, furniture or other objects in the indoor environment can also be matched using the proposed 2EC features.
Appendix A

Input Image Pairs

This appendix represents fifteen image pairs which were used in our experiments in Section 4.2. For each pair of images, fundamental matrix was computed using the match correspondences which were established using 2EC features. Three epipolar lines, generated using the fundamental matrix, are shown in each figure. Red dots in each figure are the points that corresponds to the epipolar lines.

Figure A.1: Image pair 90_1: the epipolar lines are generated from the matching results.

Figure A.2: Image pair 90_2: the epipolar lines are generated from the matching results.
APPENDIX A. INPUT IMAGE PAIRS

Figure A.3: Image pair 90.3: the epipolar lines are generated from the matching results.

Figure A.4: Image pair 90.4: the epipolar lines are generated from the matching results.

Figure A.5: Image pair 90.5: the epipolar lines are generated from the matching results.

Figure A.6: Image pair 180.1: the epipolar lines are generated from the matching results.
APPENDIX A. INPUT IMAGE PAIRS

Figure A.7: Image pair 180.2: the epipolar lines are generated from the matching results.

Figure A.8: Image pair 180.3: the epipolar lines are generated from the matching results.

Figure A.9: Image pair 180.4: the epipolar lines are generated from the matching results.

Figure A.10: Image pair 180.5: the epipolar lines are generated from the matching results.
Figure A.11: Image pair 270_1: the epipolar lines are generated from the matching results.

Figure A.12: Image pair 270_2: the epipolar lines are generated from the matching results.

Figure A.13: Image pair 270_3: the epipolar lines are generated from the matching results.

Figure A.14: Image pair 270_4: the epipolar lines are generated from the matching results.
Figure A.15: Image pair 270.5: the epipolar lines are generated from the matching results.
Appendix B

Portability Issues

All the parameters that were mentioned in Chapter 3 are listed below. Selecting proper values for these parameters was crucial for obtaining best matching results. Even though most of the parameters used in our algorithm were determined experimentally based on the property of our aerial images, there were certain rules to follow when choosing values for them. In the following tables, the first column provides the symbol of each parameter as was used in the main body of this thesis. The column of Description briefly explains the purpose of each parameter. The column of Value Determination explains how the parameter was set. The effect caused by increasing or decreasing each parameter is also discussed. The value of each parameter as set in our system is shown in the fourth column. All the values of parameters were remained unchanged for all the experiments in Chapter 5. Besides, it must be noticed that all the values given here were used for the down-sampled images.
Table B.1: Parameters for the line detection and linking algorithms (Sections 3.2.1 and 3.2.2).

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
<th>Value Determination</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{lsr}$</td>
<td>This parameter is the minimum area allowed to form a line support region in line detection algorithm [43].</td>
<td>As many as line segments should be detected in this step, although many incorrect lines may be included. In order to keep a balance between the above rule and the computational cost issue, the value of 10 pixels was determined experimentally.</td>
<td>10 pixels</td>
</tr>
<tr>
<td>$d_l$</td>
<td>Lateral threshold used to determine whether to link two lines.</td>
<td>The lines should be very close in order to be linked. The effect of increasing or decreasing the value was discussed in Section 3.2.2.</td>
<td>1.5 pixel</td>
</tr>
<tr>
<td>$t_{a}$</td>
<td>Angle threshold used to determine whether two lines are collinear and thus should be linked or not.</td>
<td>The two lines should have very similar orientation in order to be linked. The effect of increasing or decreasing the value was discussed in Section 3.2.2.</td>
<td>10 degrees</td>
</tr>
<tr>
<td>$t_{overlap}$</td>
<td>This parameter is the maximum overlap ratio for two lines to be linked.</td>
<td>The value was set based on recommendations in [44]</td>
<td>0.15</td>
</tr>
<tr>
<td>$t_{underlap}$</td>
<td>This parameter is the maximum underlap ratio for two lines to be linked.</td>
<td>The value was set based on recommendations in [44]</td>
<td>0.15</td>
</tr>
</tbody>
</table>
Table B.2: Parameter for edge map creation (Section 3.2.3).

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
<th>Value Determination</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{iso}$</td>
<td>This parameter is used to remove small blobs in the edge map.</td>
<td>As many as isolated blobs that correspond to protuberant objects on the rooftops should be removed. Meanwhile, we should also avoid deleting useful part of the edge map. The value for the parameter was determined experimentally by finding that all blobs under 100 pixels on the edge map are redundant. This value represents areas of about $2.25m^2$.</td>
<td>100 pixels</td>
</tr>
</tbody>
</table>

Table B.3: Parameters for 2EC feature extraction I (Section 3.2.4).

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
<th>Value Determination</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{pline}$</td>
<td>This parameter is used to remove partial lines.</td>
<td>This value was determined experimentally. By comparing the line detection results with the edge map, the value of 80% was determined to remove those lines that are not correctly represent the rooftop boundaries. Increasing this value will result in less line segments for the successive procedure and less processing time, but some correct line segments may also be removed. Decreasing this value will lead to more line segments and longer processing time.</td>
<td>80%</td>
</tr>
<tr>
<td>$l_{E_{max}}$</td>
<td>This threshold is the maximum length of extension for a line segment.</td>
<td>The value of this parameter was chosen to be half of the longest building edge. In our images, half of the longest building edge was about 150 pixels, which is about 45 meters in the 3D world.</td>
<td>150 pixels</td>
</tr>
<tr>
<td>$l_{E_{min}}$</td>
<td>This threshold is the maximum length of continued extension for a line segment.</td>
<td>This value was determined experimentally by measuring the maximum gap between the endpoints of line segments and the rooftop vertices that they might reach.</td>
<td>10 pixels</td>
</tr>
</tbody>
</table>
Table B.4: Parameters for 2EC feature extraction II (Section 3.2.4).

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
<th>Value Determination</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{ma}$</td>
<td>This parameter is the maximum orientation difference for two lines to be merged.</td>
<td>The value of this parameter should be kept very small in order to avoid merging incorrectly. The value of 5 degrees was determined experimentally.</td>
<td>5 degrees</td>
</tr>
<tr>
<td>$d_{ml}$</td>
<td>This parameter is the maximum lateral distance for two lines to be merged.</td>
<td>The value of this parameter should be kept very small in order to avoid merging incorrectly. The value of 1.5 pixel was determined experimentally, which equivalent to 0.45 meters in the 3D world</td>
<td>1.5 pixel</td>
</tr>
<tr>
<td>$\theta_{min}$</td>
<td>This parameter is the minimum cross angle for two lines to be intersected.</td>
<td>The value of this parameter was determined by finding the smallest angle of the rooftop corners in our images.</td>
<td>20 degrees</td>
</tr>
<tr>
<td>$\theta_{max}$</td>
<td>This threshold is the maximum cross angle for two lines to be intersected.</td>
<td>The value of this parameter was determined by finding the largest angle of the rooftop corners in our images.</td>
<td>160 degrees</td>
</tr>
<tr>
<td>$t_{closeinter}$</td>
<td>This threshold is used to remove short line segments.</td>
<td>The value was determined based on the fact that the length of a rooftop edge is unlikely to be shorter than 1.5 meters in the 3D world. 1.5 meters correspond to 5 pixels in our images.</td>
<td>5 pixels</td>
</tr>
<tr>
<td>$t_{short}$</td>
<td>This threshold is used to remove short line segments that are connected together forming a chain.</td>
<td>Considering that if short lines connect together forming a chain, they are unlikely to represent meaningful rooftop structures. The value was determined experimentally by measuring the incorrect line chains detected in our images.</td>
<td>10 pixels</td>
</tr>
<tr>
<td>$t_{fex}$</td>
<td>This parameter is used to extract 2EC features with longer line segments.</td>
<td>This parameter was determined based on the fact that building rooftops with their edges longer than 4.5 meters (in the 3D world) are large and distinctive enough to be identified uniquely using 2EC features. 4.5 meters correspond to 15 pixels in our images. Besides, most of the rooftop edges are longer than 15 pixels in our images.</td>
<td>15 pixels</td>
</tr>
</tbody>
</table>
Table B.5: Parameters for search space creation (Section 3.3.4).

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
<th>Value Determination</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{square}$</td>
<td>This parameter is the maximum offset that a predicted feature location could have from its ground truth location.</td>
<td>This value was experimentally determined based on the test results given in Section 3.3.3.2.</td>
<td>500 pixels</td>
</tr>
<tr>
<td>$s_{window}$</td>
<td>This parameter is the maximum offset that a predicted feature location could have from its epipolar line.</td>
<td>This value was experimentally determined based on the test results given in Section 3.3.3.2.</td>
<td>150 pixels</td>
</tr>
</tbody>
</table>

Table B.6: Parameters for filtering candidate features based on feature shape (Section 3.3.5).

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
<th>Value Determination</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{len}$</td>
<td>The threshold is the maximum length difference allowed for a 2EC feature to become a candidate match.</td>
<td>The value was determined based on the test results of the prediction accuracy in Section 3.3.3.2.</td>
<td>10%</td>
</tr>
<tr>
<td>$T_{angle}$</td>
<td>The threshold is the maximum orientation difference allowed for a 2EC feature to become a candidate match.</td>
<td>The value was determined based on the test results of the prediction accuracy in Section 3.3.3.2.</td>
<td>5 degrees</td>
</tr>
</tbody>
</table>

Table B.7: Parameters for filtering candidate features based on image correlation (Section 3.3.6).

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
<th>Value Determination</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{corr}$</td>
<td>The threshold is the minimum correlation score that two 2EC features are considered as a match.</td>
<td>This value was chosen considering that the feature prediction in Section 3.3.3.1 may contain certain error due to the imprecise camera parameters given in the metadata. The value was determined experimentally, and increasing the value may result in less correspondences, but higher correct rate.</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Table B.8: Parameters for removal of the outliers (Section 3.3.7).

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
<th>Value Determination</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_{\text{tre}} )</td>
<td>This parameter is the maximum residual error allowed for terminating the iterations of the outliers removal.</td>
<td>This value was chosen considering the fact that the ground is not a plane because of the height variations of buildings and terrain. Increasing this value may result in more matches that locate on higher building rooftops, while decreasing the value may result in less matches, which almost lie on the same plane.</td>
<td>30 pixels</td>
</tr>
<tr>
<td>( n )</td>
<td>This parameter is the minimum number of match correspondences that left after the outliers removal.</td>
<td>To compute a projective transformation between two sets of point correspondences, at least 8 pairs of points are needed. The value 20 was chosen, because the fundamental matrix computation requires more match correspondences to have less bias. Increasing this value may result in more matches but low correct matching rate. Decreasing this value may result in less match correspondences and less accuracy when computing fundamental matrix.</td>
<td>20</td>
</tr>
</tbody>
</table>
Bibliography


