AN AUTOMATIC ALGORITHM FOR TEXTURED DIGITAL ELEVATION MODEL FORMATION USING AERIAL TEXEL SWATHS

by

Taylor C. Bybee

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Dr. Scott E. Budge Major Professor	Dr. Jacob H. Gunther Committee Member
Dr. Don L. Cripps Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2015

Copyright © Taylor C. Bybee 2015

All Rights Reserved

Abstract

An Automatic Algorithm for Textured Digital Elevation Model Formation using Aerial Texel Swaths

by

Taylor C. Bybee, Master of Science

Utah State University, 2015

Major Professor: Dr. Scott E. Budge
Department: Electrical and Computer Engineering

Textured digital elevation models (TDEMs) have valuable use in precision agriculture, situational awareness, and disaster response. However, scientific-quality models are expensive to obtain using conventional aircraft-based methods. Photogrammetry-based techniques have no direct measurements, and thus has uncertainty in the reconstruction. The concept of a texel camera, which has both aerial imagery and ladar measurements from an inexpensive small UAV, can be used to combine the two methods.

A texel camera fuses calibrated ladar measurements and electro-optical imagery upon simultaneous capture, creating a texel image. This eliminates the problem of fusing the data in a post-processing step and enables both 2D- and 3D-image registration techniques to be used. A texel camera outputs texel swaths during a UAV flight. A swath consists of an aerial image that is calibrated to associated depth measurements. This thesis describes an automatic algorithm for registering these texel swaths into a TDEM.

The algorithm involves image processing, 3D geometry, and nonlinear optimization processes. The algorithm is seeded with a coarse estimate of the position and attitude of each texel swath capture, obtained using an on-board navigation system. Analysis of several

data sets registered using this algorithm is shown. This method enables an inexpensive alternative to obtaining high quality textured 3D landscapes.

(104 pages)

Public Abstract

An Automatic Algorithm for Textured Digital Elevation Model Formation using Aerial Texel Swaths

by

Taylor C. Bybee, Master of Science

Utah State University, 2015

Major Professor: Dr. Scott E. Budge
Department: Electrical and Computer Engineering

The process of creating a 3D terrain map of an area is a challenging, computationally-intensive task. There are two main camps of established methods doing this, with varying degrees of accuracy and cost. Using the established methods, there is a trade-off between accuracy and cost. The first method involves using many aerial images to detect disparity between points in the images. This is a difficult task as it requires a lot of computer processing with varying degrees of reliability. In addition, this method does not make any direct distance measurements. Secondly, using high-precision and high-cost lasers and positioning equipment, measurements can be taken with a high degree of accuracy. The cost of this is not insignificant.

In order to combine the two methods, this thesis describes an automatic process which is a hybrid mixture of the afore-mentioned methods. Utah State University has developed the concept of a texel camera, which takes both laser measurements and a digital imagery together in a calibrated fashion. Using both the digital image and laser measurements, the process described in this thesis enable the formation of a 3D terrain map with almost no human intervention.

"Be sure your wisest words are those you do not say."
Robert W. Service

...

"In reality, we are all travelers — even explorers of mortality."
Thomas S. Monson

Acknowledgments

I express gratitude to my major professor, Dr. Scott Budge, for his help and guidance throughout this process. I also acknowledge my family for their support during this busy time as a student, as well as my friends who have been mindful of the effort I've made. I am also grateful for my committee members, other faculty members, and university staff who have been supportive of me as a student.

I am thankful for my fellow students in the Center for Advanced Imaging Ladar: Xuan Xie, Cody Killpack, Seth Andrews, Bikalpa Khatiwada, and Serena Makin. They've given valuable insight into my research in our group's student research meetings. I also express appreciation to Robbie Schaap who volunteered his time to do the mechanical computer design for the texel camera bracket. I have gratitude for the Utah Water Research Laboratory's group, Aggie Air, for their insight into aerial vehicles, especially Nathan Hoffer. I express appreciation for Dr. Xiaojun Qi in the Computer Science Department for the input on image processing and orthorectification. I am thankful for Heidi Harper in the ECE store for resources and tools needed for the construction of the texel camera.

I acknowledge this research has been funded by the Space Dynamics Laboratory at Utah State University. Much of the research presented in this document, including some plots, figures, and phraseology, has been published in SPIE conference 9465 proceedings.

Taylor C. Bybee

Contents

Page

Abstract . . . . . **iii**

Public Abstract . . . . . **v**

Acknowledgments . . . . . **vii**

List of Tables . . . . . **x**

List of Figures . . . . . **xi**

Acronyms . . . . . **xiii**

1 Introduction **1**
1.1 Previous Work 3
1.2 Contribution 4

2 Texel Camera Basics, Camera Geometry, and Image Processing Techniques **6**
2.1 Texel Cameras: System Overview, History, and Definitions 6
2.2 Optical System Design and Calibration 7
2.2.1 Calibration 9
2.3 Camera Location and Attitude Convention 13
2.3.1 Quaternions 13
2.3.2 Camera Location and Attitude as a Matrix 14
2.3.3 Moving a Point Into Another Coordinate System 15
2.4 Normalized Image Plane Projections 16
2.4.1 Projection into the Normalized Image Plane 17
2.4.2 Projection and Range: An Alternative to Cartesian Coordinates . . 18
2.4.3 Normalized Image Plane and Column-Row Coordinates 20
2.5 Homography 22
2.5.1 Types of Homography 23
2.5.2 Finding a Homography using Matching Points 25
2.5.3 Random Sampling Consensus 27
2.6 Harris Feature Points 27
2.7 Normalized Cross-Correlation 29
2.8 Fundamental Matrix and Epipolar Geometry 31
2.8.1 Computing the Fundamental Matrix 33
2.8.2 Recovering Rotation and Translation from the Fundamental Matrix 34
2.8.3 Applications of the Fundamental Matrix 37
2.9 Conclusion 38

3 Triangulation of Texel Swaths . . . 39
3.1 Swath Registration and 3D Reconstruction Problem . . . 39
3.2 Triangulation of 3D Points . . . 40
3.3 Algorithm for Finding Projection Points . . . 45
3.3.1 Finding an Initial Homography Estimate from Camera Position and Attitude . . . 46
3.3.2 Finding Projection Points in Non-Adjacent Images . . . 49
3.4 Bundle Adjustment Optimization . . . 50
3.4.1 Sparse Levenberg-Marquardt Algorithm . . . 51
3.4.2 Incremental Optimization Approach . . . 56
3.4.3 Seeding the Optimization . . . 56
3.5 Textured Digital Elevation Model Creation . . . 58

4 Experimental Results . . . 61
4.1 Texel Swath Acquisition . . . 61
4.2 Image Processing Analysis . . . 62
4.2.1 Harris Feature Point-Finding . . . 63
4.2.2 Fundamental Matrix Calculation . . . 64
4.2.3 Finding Projection Points . . . 66
4.3 Swath Registration Results . . . 68
4.3.1 Metric for Analysis . . . 70
4.3.2 Data Set Analysis . . . 70
4.3.3 Comparison to Photogrammetry . . . 79
4.4 Discussion . . . 79

5 Conclusion and Future Work . . . 82
5.1 Conclusion . . . 82
5.2 Future Work . . . 83

List of Tables

Table Page

4.1 Error for Various EO Widths for Level Flight Data Set 71

4.2 Error for Various EO Widths for Turbulent Flight Data Set 74

4.3 Error for Various EO Widths for Turn Flight Data Set 77

List of Figures

Figure Page

1.1 Gathering data from a small UAV with a texel camera. 2

1.2 Texel swath concept. 3

2.1 Example texel image showing the author at his desk. 7

2.2 Depiction illustrating the pinhole camera model. 8

2.3 Transmission and reflection curves for a cold mirror. 9

2.4 Ray diagram of a handheld texel camera. 10

2.5 Setup for finding the COP of the depth camera and adjusting the cold mirror and EO camera locations. 11

2.6 Axis-angle representation of a rotation. 13

2.7 Normalized image plane illustrations. 17

2.8 A diagram showing the projection of a 3D point onto the normalized image plane. 18

2.9 Projection-range coordinate system. 21

2.10 A comparison of image coordinate systems. 22

2.11 Mapping corresponding points using a homography. 23

2.12 Comparing two patches in two images using NCC. 30

2.13 Two-camera epipolar geometry. 32

2.14 Finding the best translation given isotropic noise on the initial guess. 37

3.1 Projection of a 3D point into image swaths. 41

3.2 Sources of error in the system. 44

3.3 Finding projection points in adjacent swaths. 47

3.4 Finding a homography from 3D points and camera information. 50

3.5 Types of system bundle adjustment. 57

3.6 Showing the collinearity problem. 58

3.7 Relationship between final texture (left) and a given texel swath (right). . . 59

4.1 Data set acquisition. 62

4.2 The dimension of the image being trimmed about the ladar shots. 62

4.3 An image with Harris feature points highlighted in red. 63

4.4 Putative correspondences and associated epipolar lines using the fundamental matrix derived from the measured R and $\mathbf{t}$. 64

4.5 Matching points and corresponding epipolar lines from the fundamental matrix found using corresponding points 66

4.6 Ambiguity of small rotations and translations. 67

4.7 Image points projected from ladar points found using correlation in adjacent swaths. 69

4.8 Smooth flight data set registration. 72

4.9 Smooth flight full-frame reference texel image. 73

4.10 Roll profile for the turbulent flight data set. 74

4.11 Turbulent flight data set registration 75

4.12 Flight pattern for the turn data set. 77

4.13 Flight with turn data set registration. 78

4.14 Comparison of photogrammetry to texel swath optimization. 80

4.15 Side comparison of photogrammetry to texel swath optimization. 81

Acronyms

AHRS	Attitude and Heading Reference System
CAIL	Center for Advanced Imaging Ladar
COP	Center Of Projection (i.e., camera center)
DEM	Digital Elevation Model
EO	Electro-Optical
FFT	Fast Fourier Transform
FOV	Field Of View
GCP	Ground Control Point
GPS	Global Position System
INS	Inertial Navigation System
LMA	Levenberg-Marquardt Algorithm
LUT	Look-Up Table
NCC	Normalized Cross-Correlation
NIR	Near Infra-Red
RANSAC	RANdom SAmpling and Consensus
RGB	Red-Green-Blue (a colorspace)
SPIE	Society of Photographic Instrumentation Engineers
SVD	Singular Value Decomposition
TDEM	Textured Digital Elevation Model
UAV	Unmanned Aerial Vehicle
USU	Utah State University
YIQ	luminance-intensity-chrominance (a colorspace)

Chapter 1

Introduction

Three-dimensional representations of a landscape created from small unmanned aerial vehicles are not only intriguing but valuable in the areas of precision agriculture, disaster response [1], watershed evaluation, ecology [2], forestry [3], archaeological/historical records [4], and defense. Value in 3D landscape models increase significantly when there is a *texture*, or image, overlaid on the 3D information. This enables, for instance, a farmer to not only monitor erosion in the field, but also be able to view any invasive species present in the field. On a more serious scale, it enables the military to identify the difference between an opposing army truck and a school bus (i.e. similar shapes, but different coloring). The value of the 3D information combined with digital imagery is skyrocketing in a variety of applications.

The creation of textured 3D landscapes, or rather, textured digital elevation models (TDEMs) is a busy area of research. There are many well-established methods for creating 3D representations of landscapes, but each has its strengths and drawbacks. These methods can be categorized into two camps: photogrammetry and range sensing. Photogrammetry uses aerial images to triangulate elevation in a scene. Range sensing measures the distances to the ground from sensors in an aerial vehicle. The most common type of range measurement is ladar (a portmanteau of LAser and raDAR) which uses light, or, more specifically, lasers, to measure distance.

These two methods contrast two aspects of gathering information. A comparison is given by Baltsavias [5]. The inherent problem with photogrammetry is that no direct measurements are made; depth information is *inferred*. The two inherent problems, one practical and the other theoretical, with range sensing are (1) cost and (2) identifying pixels in overlaid imagery to measured 3D points. Thus, there is a trade-off between accuracy

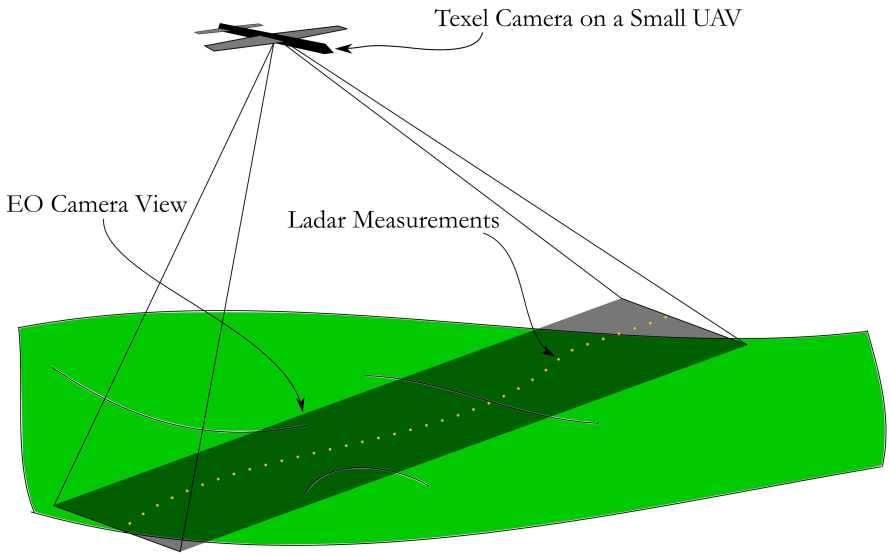


Fig. 1.1: Gathering data from a small UAV with a texel camera.

(range sensing) and low-cost (photogrammetry), as expressed by Conte et al [6]. This thesis exploits the idea of using the best of both methods to obtain a TDEM from a small unmanned aerial vehicle (UAV) and relatively inexpensive ladar system using the concept of a texel camera. An illustration showing this data collection is shown in Fig. 1.1.

A texel camera captures both a digital image and ladar measurements simultaneously, and calibrates the information upon capture. This solves the problem of mapping 3D points to digital imagery in a post-processing step. Due to the nature of many aerial ladar systems, only a small number of "strips" of range information is gathered at a time; in other words, due to the moving nature of the vehicle, the system cannot scan the entire scene from one perspective in space. Because only a small number of strips of ladar information are captured and to reduce throughput, only the digital image surrounding the strip needs to be captured. This kind of texel image is called a texel swath. This is shown in Fig. 1.2.

This thesis describes the process of combining texel swaths (with their respective digital images and range measurements) into a single textured digital elevation model. The resultant model is accurate because of the measured range measurements, and its texture is

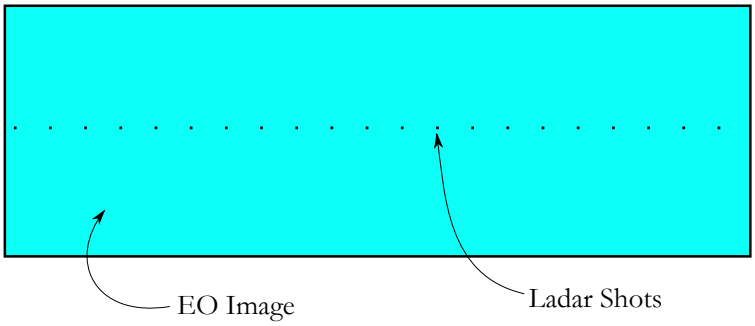


Fig. 1.2: Texel swath concept.

reliable because it is calibrated to the range measurements. The trade-off between cost and accuracy is circumvented due to the calibrated and simultaneous capture of aerial imagery and range measurements.

1.1 Previous Work

Photogrammetry

The concepts in photogrammetry have existed for decades [7], and have evolved from film-based imagery to digital imagery [8]. Photogrammetry allows images taken from an aerial vehicle to be triangulated and reconstructed into a 3D scene. This is analogous to stereo vision, but with many more points of view. A survey of image registration methods is given by Zitova [9]. Typically, points are identified from image-to-image, matched [10], then triangulated. This can be done manually or automatically [11]. If done manually, the risk of a human error is present. Thus, robust algorithms for completely automatic methods are desired [12–15]. Rather than create an TDEM, other methods focus on extracting 3D objects such as buildings [16–18]. Most methods require calibrated cameras and large aerial vehicles, but other methods do not [19, 20].

Because of 3D-to-2D projections inherent in taking a photograph, scale information is lost. This can be offset to some degree using ground control points (GCPs) or a 2D base map [21]. Other methods include matching aerial imagery to existing 3D information [22].

Range Sensing

Range sensing measurements are only as accurate as the knowledge of the point from which they are measured [23], and, thus, high-precision inertial navigation systems (INS) and global positioning systems (GPS) must be used to ensure the resulting 3D measurements are accurate. These position and attitude systems are expensive. The ladar equipment is also expensive. However, with the expense comes great accuracy in the resulting landscape models. An aerial digital image can be overlaid on top of the 3D model in post-processing so a TDEM is formed [24–28], which is not a trivial process. There has been other work done integrating range sensing and photogrammetry [29–33], but still require a post-processing step.

1.2 Contribution

Rather than registering digital images to a point cloud, this thesis describes registering texel swaths, which are bundles of imagery and points, to one another. In other words, both the point cloud and digital image locations on that point cloud are adjusted during the registration.

Because the digital image of a texel swath has one dimension that is significantly larger than the other dimension, photogrammetry techniques may fail because there are not a large number of pixels in the overlapping regions, and good matching points may not be automatically selected. This is mitigated using the techniques presented in this research. Additionally, a cost function which incorporates the 3D point measurements (as well as point-matching techniques on the images) is used to minimize the system error in the 3D reconstruction. The measured 3D points reduce the scale ambiguity of the image matching and help recover the relative location and pose of each camera, enabling 3D points to be accurate. Because of these technical advantages, this research is a valuable contribution in creating TDEMs.

This low-cost yet accurate ability to create a TDEM enables a farmer to have a field surveyed multiple times during a single season for erosion control, environmental planning, and evaluation of watering needs. It allows for a watershed to be mapped several times

during the course of the annual runoff to monitor changes in river bank structure. It allows for quick assessment of a battlefield or rural location. It has valuable use because it is a low-cost alternative to high-cost methods.

Chapter 2 outlines the concept of the texel camera as well as the basics of camera geometry and image processing. Chapter 3 explains the mathematical basis for creating a TDEM from texel swaths and the optimization process. Chapter 4 describes the data collection and experimental registration results. Finally, Chapter 5 offers a conclusion.

Chapter 2

Texel Camera Basics, Camera Geometry, and Image Processing Techniques

This chapter introduces the concept (Section 2.1), design, and calibration (Section 2.2) of the texel camera used for the research presented in the document. It also introduces basic 3D rotations for representing a camera in 3D space (Section 2.3), as well as projections from 3D space onto the camera image plane (Section 2.4). Furthermore, image processing concepts such as homography (Section 2.5), Harris features points (Section 2.6), normalized cross-correlation (Section 2.7), and epipolar geometry (Section 2.8) are introduced.

2.1 Texel Cameras: System Overview, History, and Definitions

The Center for Advanced Imaging Ladar (CAIL) at Utah State University (USU) has developed the concept of the texel camera [34]. This camera captures both color and depth information simultaneously, creating a 2.5D image (a 3D image captured from a single point of view), called a texel image. This information is calibrated upon capture.

A texel image is a collection of data containing range measurements which are mapped through a calibration process to an electro-optical (EO) image, more commonly known as a digital image. That is, each depth measurement is assigned a calibrated location on the EO image. The EO image has a higher resolution than the depth measurements; there is texture information between measured points. An example texel image of the author is shown in Fig. 2.1.

A texel camera is a device used to capture texel images. The device used in this research is a second-generation handheld texel camera. It incorporates a depth sensor as well as an EO sensor arranged in such a way that the sensors have a common center of projection (COP). These sensors (cameras) are calibrated to produce a texel image.

Fig. 2.1: Example texel image showing the author at his desk.

2.2 Optical System Design and Calibration

The second-generation handheld texel camera consists of two co-boresighted cameras: a PMD flash ladar camera (CamCube 2.0) and a Micron/Aptina EO camera. The pinhole camera model is assumed for each camera. Also mounted on the device is a Vector-Nav VN-100 Attitude and Heading Reference System (AHRS) to record camera orientation. In this document, the pinhole is also referred to as the camera center of projection (COP), or merely the camera center. The pinhole model is shown in Fig. 2.2. The cameras must be calibrated to one another.

To ensure the depth and EO cameras are co-boresighted, their respective COPs must be located at the same point in space. This ensures that the cameras see the same scene from the same perspective; i.e., there is no parallax between the cameras.

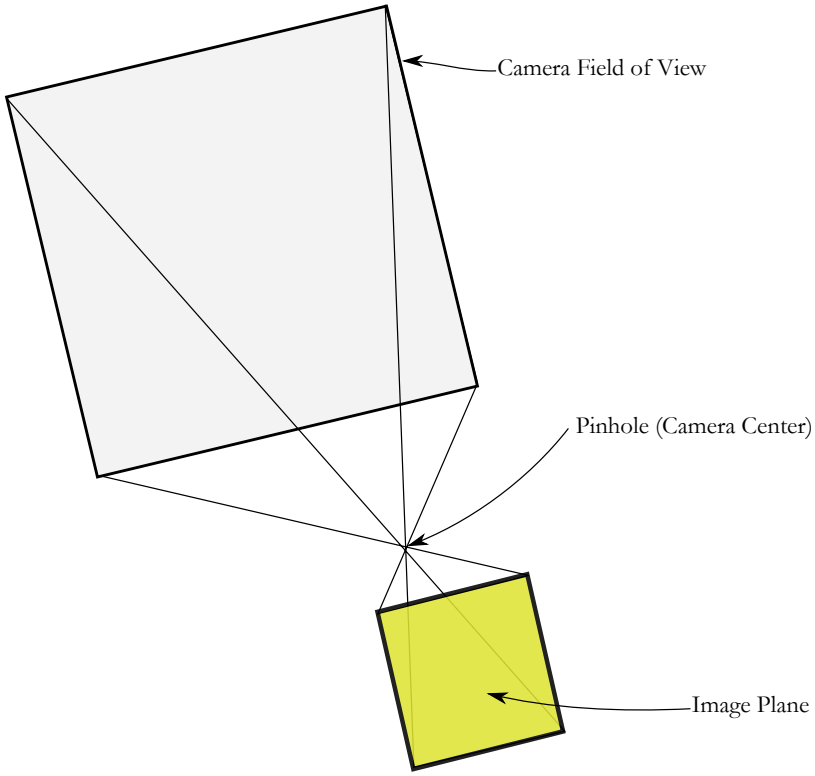


Fig. 2.2: Depiction illustrating the pinhole camera model. All rays are assumed to pass through a single point, the camera center.

The depth camera operates at a wavelength of 870 nm, in the near-infrared (NIR) range, while the EO camera operates at normal optical wavelengths (approximately 400 to 700 nm). Because the cameras operate at different wavelengths, their sensors can be co-boresighted using a type of beam-splitter called a cold mirror. A cold mirror transmits NIR wavelengths and reflects optical wavelengths. The cameras can then be placed at different locations while viewing the same scene from the same perspective. The cold mirror transmission curve is shown in Fig. 2.3, courtesy of Edmund Optics [35].

The depth and EO cameras can then be aligned such that there is no parallax between their images. There is a small amount of refraction in the cold mirror as the NIR rays pass through it. There are also lens aberrations and other imperfections in the optical system, most notably the barrel distortion of the EO camera. Most of these effects can

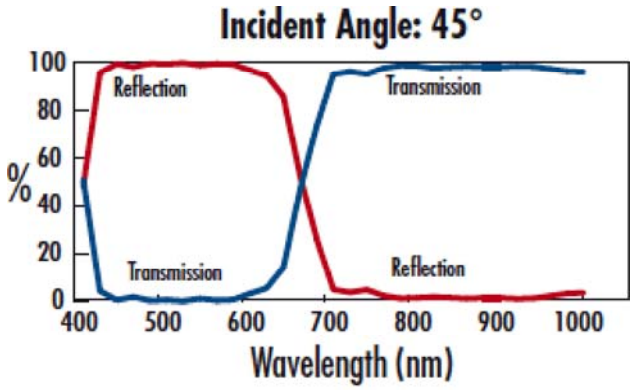


Fig. 2.3: Transmission and reflection curves for a cold mirror.

be compensated for in a geometric calibration of the camera. A ray diagram of the texel camera is shown in Fig. 2.4.

2.2.1 Calibration

Budge and Badamikar [36] describe a calibration process for the first-generation handheld texel camera. It involves three steps:

1. **Camera Alignment** This calibration step removes the parallax between the depth and EO cameras.

2. **Geometric Calibration** This step removes the lens distortions and maps the depth camera measurements to the EO image.

3. **Depth Calibration** This step calibrates the depth measurements correcting for flat-field, ladar range-intensity (wiggle error), and the origin of the measurements.

The calibration for the second-generation texel camera follows the first two steps closely. However, for the depth calibration, Budge and Badamikar assume access to the raw, uncorrected depth measurements from the depth sensor. The depth sensor used in the second-generation texel camera, the PMD CamCube 2.0, automatically corrects for wiggle error, and is proprietary to the camera manufacturer. This necessitates a modification to the cited

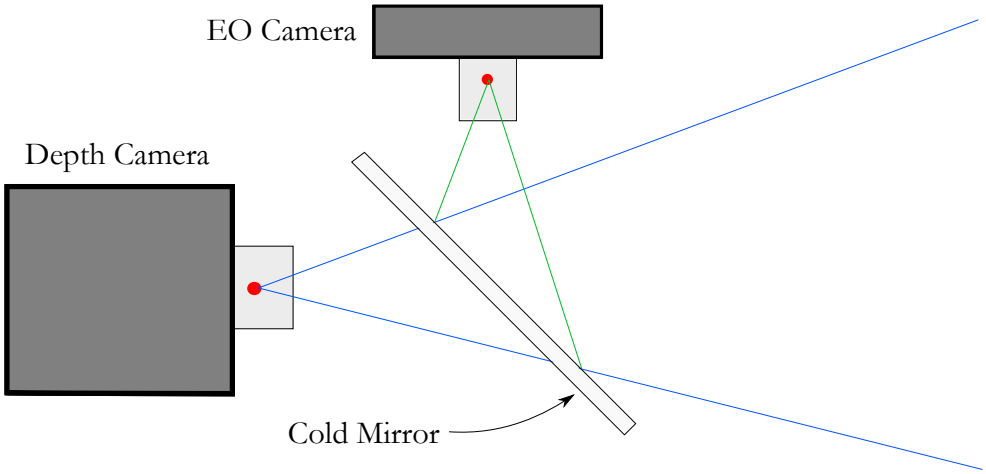


Fig. 2.4: Ray diagram of a handheld texel camera. The red dots show the camera COPs. The blue and green rays represent the edges of the field of view (FOV) of each camera.

calibration procedure. Details about the process and mapping equations used in Steps 1 and 2 are given by Budge and Badamikar [36].

Camera Alignment

The camera alignment involves iteratively adjusting the location of the cold mirror and, if necessary, the relative locations of the depth and EO cameras. There are adjustment screws which easily allow this to happen.

When a camera is rotated about its COP, there is no parallax introduced during the rotation. A panoramic camera mount was used to find the COP of the depth camera. The EO camera location or the orientation of the cold mirror is adjusted so the EO camera has its COP at the "same location" as the depth camera.

To determine if there is parallax introduced during the rotation on the panoramic mount, two paper triangles are placed at different distances from the camera. When the camera is looking directly at these triangles, they are positioned such that two vertices appear to be touching. If there is parallax, the triangles will no longer appear to be touching when the camera is rotated. If there is no parallax during the rotation, the triangles will not move with respect to one another in the image, and the point about the rotation occurred is considered the COP. This setup is shown in Fig. 2.5. Once the COP is found, the line of

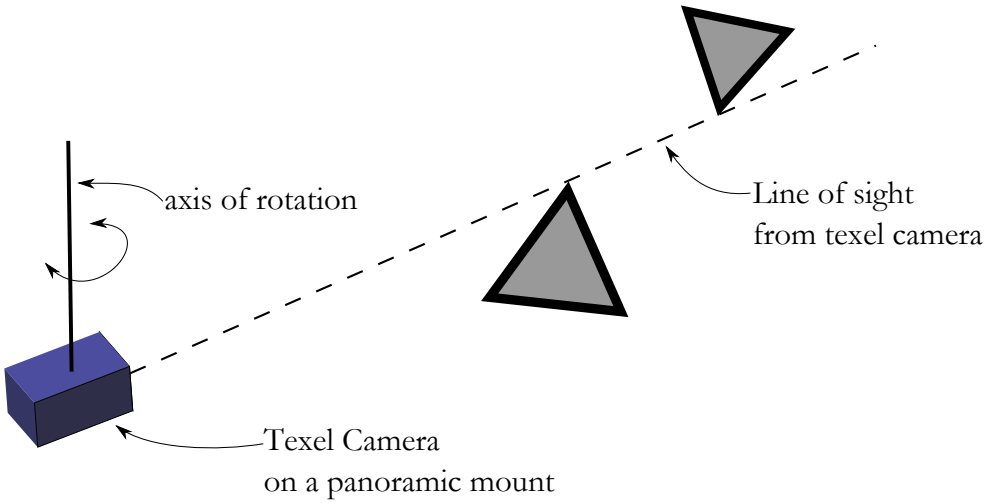


Fig. 2.5: Setup for finding the COP of the depth camera and adjusting the cold mirror and EO camera locations.

sight will pass through both triangle vertices no matter the rotation about the COP. This process is done both for both horizontal and vertical axes of rotation.

Geometric Calibration

Once the cameras are known to be co-boresighted, the depth camera pixels are mapped to the appropriate positions in the EO image. Concurrently, the images are calibrated to remove any lens distortion. This is done by taking many images of a checkerboard pattern from different perspectives using both cameras. In addition to 3D points, the depth camera also returns an intensity image. These intensity images of the checkerboard pattern are analyzed in the MATLAB Camera Calibration toolbox [37] to calibrate the depth camera (i.e. remove nonlinear distortion). Once the depth camera is calibrated, corresponding points in the EO image are identified. A pixel-by-pixel mapping is defined between the two images. At this point both images are calibrated to one another. Each depth pixel has a calibrated location in the EO image.

Range Calibration

The range calibration is the most involved calibration. Many things affect the final range measurement including individual pixel variations, intensity of the return signal, and

the location of the measurement origin. The measurement origin is discussed by Budge and Badamikar [36]. The PMD camera range measurements have a standard deviation of about four millimeters for areas of the image with good intensity returns.

Flat-field Calibration

When taking depth measurements of a flat wall of constant intensity, it is expected that the resulting measurements reflect a plane in 3D space.

In reality each depth pixel's response may differ from one another, making the plane look noisy. However, much of this error is not randomly distributed but structured. That is, each pixel has a consistent error in its measurement. For example, one pixel may consistently make measurements that are one millimeter closer than the true value. Another pixel may make measurements that are two millimeters farther than the true value. When taking an image of a flat wall, the resulting 3D measurements will not exactly be a plane.

This is corrected by averaging many range measurements of a flat wall to remove the zero-mean random noise, leaving structured error. This error offset is assigned pixel-by-pixel. This range error is simply added to the original range measurement, pixel-by-pixel.

Intensity Calibration

Because the manufacturer of the depth camera incorporates a built-in calibration which removes wiggle error, there is no need to follow the procedure in Budge and Badamikar which calibrates out wiggle error. However, there are still errors related to the intensity return for each pixel that are unaccounted.

A flat wall at a known distance is used as a scene. Rather than having a constant color (as in flat-field calibration), colors with varying NIR intensities are placed on the wall. Many measurements are averaged together to remove the random noise. The error and intensity for each pixel is evaluated, based on error from the plane representing the wall. A polynomial is fitted to the data to describe the relationship between error and intensity. To correct for the intensity error, the error offset calculated by the polynomial relationship is simply added at each capture.

The variations in the error increase significantly for lower intensity. It is difficult to make an accurate measurement with low intensity returns.

2.3 Camera Location and Attitude Convention

This section outlines the basic convention used in the CAIL research group for camera attitude and location, and, more generally, 3D isometric rotations.

2.3.1 Quaternions

Quaternions are a 4D representation of a number $q = q_0 + q_1 i + q_2 j + q_3 k$ where $i^2 = j^2 = k^2 = ijk = -1$. It is an extension of complex numbers. They were discovered by the 19th Century physicist Sir William Rowan Hamilton, who famously carved an equation in a bridge [38]. Quaternions are usually applied in engineering and computer graphics as a way to describe 3D rotations. Just as complex numbers can be used to describe 2D rotations, quaternions can be used to describe 3D rotations. Quaternions are better (in most applications) than Euler angles because there is no gimbal lock in quaternions, and there is no need to order yaw, pitch, and roll angles. Quaternions can be best described by the axis-angle representation of a rotation in space.

Geometrically, a rotation can be viewed as having a rotation axis (vector in 3D space) and an angle θ about the axis. This is shown in Fig. 2.6.

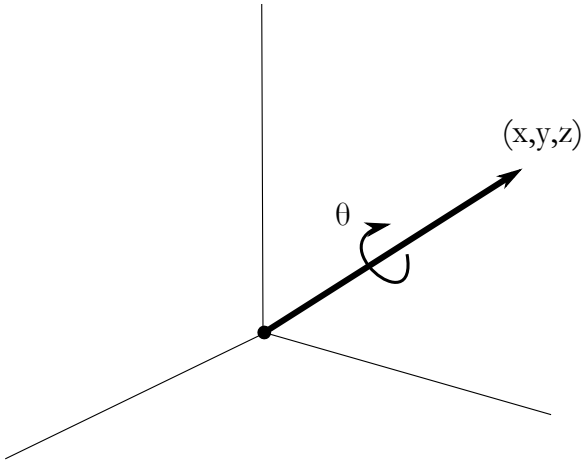


Fig. 2.6: Axis-angle representation of a rotation.

The conversion from the axis coordinates (x, y, z) and angle θ to quaternion format are simply

$$\begin{aligned} q_0 &= \cos(\tfrac{\theta}{2}) \\ q_1 &= x\sin(\tfrac{\theta}{2}) \\ q_2 &= y\sin(\tfrac{\theta}{2}) \\ q_3 &= z\sin(\tfrac{\theta}{2}). \end{aligned} \tag{2.1}$$

There is ambiguity with the axis and angle representation if the axis is in the opposite direction and the angle is negated. Quaternions are often referred to as having a scalar component (q_0) and a vector component ($[q_1, q_2, q_3]$). Some conventions place the scalar component after the vector component, but this document will use the scalar-first convention. Rotating points using quaternions and translation vectors is described in Section 2.3.3. The attitude (orientation) of an object or camera can be described as a 3D rotation from a reference orientation.

2.3.2 Camera Location and Attitude as a Matrix

The location and attitude of each camera in a set of observations can be represented by seven parameters: a normalized quaternion rotation and a 3D translation vector relative to some world coordinate system $\mathcal{O}$. Symbolically, these values can be represented as a vector. The j^{th} camera location and attitude is given by $\mathbf{a}_j = [q_{j0}, q_{j1}, q_{j2}, q_{j3}, t_{jx}, t_{jy}, t_{jz}]^T$. For this document, the scalar component of the quaternion is q_{j0}.

These values can be represented as 3×4 matrix

$$[R_j|\mathbf{t}_j] = \begin{bmatrix} 1 - \frac{2(q_{j2}^2+q_{j3}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j1}q_{j2}-q_{j0}q_{j3})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j1}q_{j3}+q_{j0}q_{j2})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & t_{jx} \\ \frac{2(q_{j1}q_{j2}+q_{j0}q_{j3})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & 1 - \frac{2(q_{j1}^2+q_{j3}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j2}q_{j3}-q_{j0}q_{j1})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & t_{jy} \\ \frac{2(q_{j1}q_{j3}-q_{j0}q_{j2})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j2}q_{j3}+q_{j0}q_{j1})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & 1 - \frac{2(q_{j2}^2+q_{j1}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & t_{jz} \end{bmatrix}, \tag{2.2}$$

which is recorded at the time of capture for each texel swath. The location of the j^{th} camera center in the world coordinate system is $\mathbf{t}_j = [t_{jx}, t_{jz}, t_{jz}]^T$. There is another

convention that is just the opposite: the translation (and rotation) is viewed as the location (and attitude) of the world coordinate system relative to the j^{th} camera coordinate system. By definition, the world coordinate system matrix is defined as

$$[R_{\mathcal{O}}|\mathbf{t}_{\mathcal{O}}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.3}$$

The matrices $[R_j|\mathbf{t}_j]$ represent the attitude and location of each capture j, relative to the world coordinate system matrix.

These matrices allow for the mapping of 3D points from one coordinate system into another when the 3D points are represented as homogeneous coordinates. In this document, the term "camera" will refer to a texel swath capture with a specific attitude and location. The matrix representation of a quaternion is used because it requires the matrix to be orthonormal.

2.3.3 Moving a Point Into Another Coordinate System

Moving a 3D point $\chi_j = [\chi_{j_x}, \chi_{j_y}, \chi_{j_z}]^T$ from the j^{th} camera coordinate system to the world coordinate system is given by the matrix multiplication

$$\begin{bmatrix} \chi_{\mathcal{O}_x} \\ \chi_{\mathcal{O}_y} \\ \chi_{\mathcal{O}_z} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 - \frac{2(q_{j2}^2+q_{j3}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j1}q_{j2}-q_{j0}q_{j3})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j1}q_{j3}+q_{j0}q_{j2})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & t_{jx} \\ \frac{2(q_{j1}q_{j2}+q_{j0}q_{j3})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & 1 - \frac{2(q_{j1}^2+q_{j3}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j2}q_{j3}-q_{j0}q_{j1})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & t_{jy} \\ \frac{2(q_{j1}q_{j3}-q_{j0}q_{j2})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j2}q_{j3}+q_{j0}q_{j1})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & 1 - \frac{2(q_{j2}^2+q_{j1}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & t_{jz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \chi_{j_x} \\ \chi_{j_y} \\ \chi_{j_z} \\ 1 \end{bmatrix}. \tag{2.4}$$

Conversely, moving a 3D point $\chi_{\mathcal{O}} = [\chi_{\mathcal{O}_x}, \chi_{\mathcal{O}_y}, \chi_{\mathcal{O}_z}]^T$ from the world coordinate system into the j^{th} camera coordinate system is given by

$$\begin{bmatrix} \chi_{j_x} \\ \chi_{j_y} \\ \chi_{j_z} \end{bmatrix} = \begin{bmatrix} 1 - \frac{2(q_{j2}^2+q_{j3}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j1}q_{j2}-q_{j0}q_{j3})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j1}q_{j3}+q_{j0}q_{j2})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} \\ \frac{2(q_{j1}q_{j2}+q_{j0}q_{j3})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & 1 - \frac{2(q_{j1}^2+q_{j3}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j2}q_{j3}-q_{j0}q_{j1})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} \\ \frac{2(q_{j1}q_{j3}-q_{j0}q_{j2})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j2}q_{j3}+q_{j0}q_{j1})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & 1 - \frac{2(q_{j2}^2+q_{j1}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} \end{bmatrix}^T \begin{bmatrix} \chi_{\mathcal{O}_x} - t_{jx} \\ \chi_{\mathcal{O}_y} - t_{jy} \\ \chi_{\mathcal{O}_z} - t_{jz} \end{bmatrix} \tag{2.5}$$

(note the transpose on the rotation matrix).

The asymmetry in the coordinate system transformations (2.4) and (2.5) is due to the use of homogeneous coordinates.

2.4 Normalized Image Plane Projections

When a camera is calibrated, all lens distortion effects are removed. This means a straight line in the scene will be a straight line on the image; it will not appear bent. The normalized image plane is a concept used to show the locations of the projections of 3D points onto the image. It can be taken either behind (as in Fig. 2.2) or in front of the camera (as in Fig. 2.7). If it is behind the camera, the images appear upside-down and reflected left-to-right.

Convention in this document will place the normalized image plane in front of the camera, perpendicular to the look-direction at a distance of one distance unit (meter in this document). Because the camera is calibrated, the coordinates on the normalized image plane can be transformed into column-row coordinates on the associated digital image. If the look-direction of the camera is the positive z-axis, then the normalized image plane is found at $z = 1$ meters away from the camera center along the principal axis of the camera.

A 3D point can be projected into the normalized image plane by using the techniques described in this section. In addition, an alternative to the Cartesian coordinate system using projection points and ranges are discussed. The relationship between the normalized image plane coordinates and column-row image indexing is described.

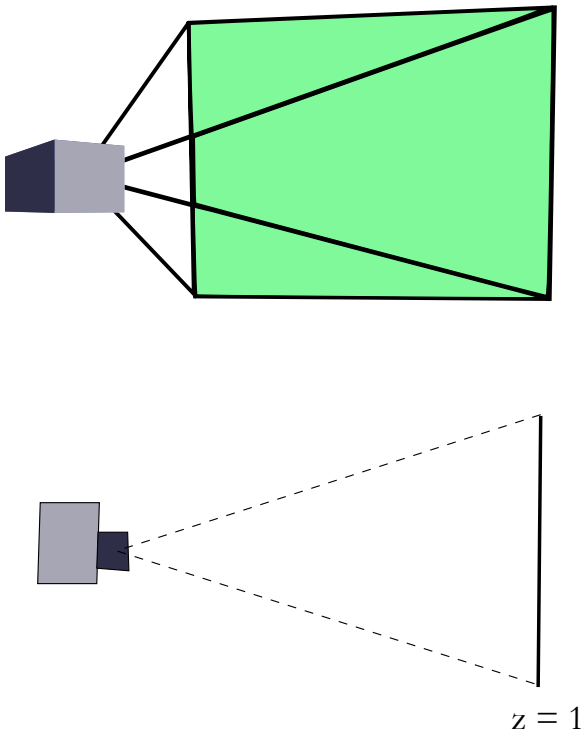


Fig. 2.7: Normalized image plane illustrations.

2.4.1 Projection into the Normalized Image Plane

A 3D point $\chi_{ij} = [\chi_{ij_x}, \chi_{ij_y}, \chi_{ij_z}]^T$ in the j^{th} coordinate system can be projected onto the j^{th} normalized image plane by

$$\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = \begin{bmatrix} \dfrac{\chi_{ij_x}}{\chi_{ij_z}} \\ \dfrac{\chi_{ij_y}}{\chi_{ij_z}} \end{bmatrix}, \tag{2.6}$$

where x_{ij} and y_{ij} represent the normalized image coordinates of the i^{th} 3D point χ_{ij} projected into the image plane j. Often a projection will be expressed in homogeneous coordinates by adding an extra dimension, usually a one in the third coordinate (this enables some nonlinear relationships to become linear). The normalized image plane in this convention is a plane that lies at $\chi_{j_z} = 1$. It is important to realize the camera projection concept follows from the pinhole model of a camera, and is derived using similar triangles. This is shown in Fig. 2.8.

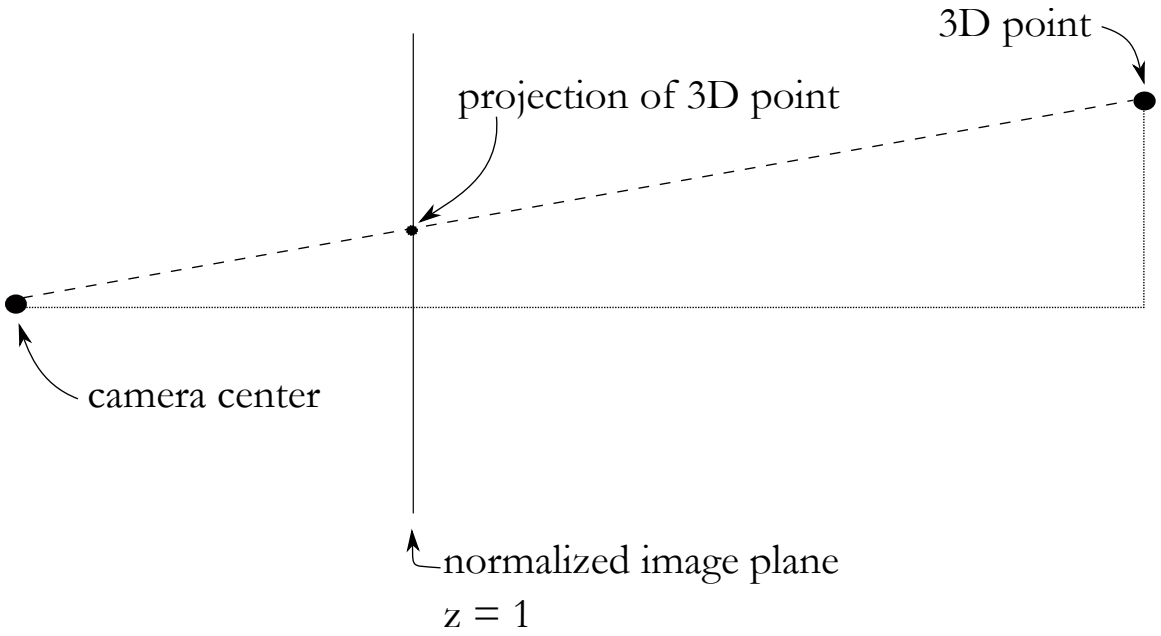


Fig. 2.8: A diagram showing the projection of a 3D point onto the normalized image plane. Any 3D point along the dotted line will project to the same point on the normalized image plane.

Information about the absolute size of an object is lost during a projection. This concept might be best explained by a natural phenomenon. From the surface of the Earth, the Sun and Moon appear to be the same size. In reality, the Sun is much larger than the Moon. However, they appear the same size because the Moon is much closer to the Earth than the Sun. The images projected onto the human retina lose 3D information about the actual sizes of the Sun and Moon, but retain only the sizes of their projections. The projection size depends upon not only the absolute size, but also the distance at which it is being projected. This is why scale is lost when 2D images are used for 3D reconstruction.

A 3D point $\chi_{i\mathcal{O}} = [\chi_{i\mathcal{O}_x}, \chi_{i\mathcal{O}_y}, \chi_{i\mathcal{O}_z}]^T$ in the world coordinate system $\mathcal{O}$ can be projected onto the k^{th} image plane by first moving the point into the k^{th} coordinate system using (2.5), then projecting onto the image plane as described in (2.6).

2.4.2 Projection and Range: An Alternative to Cartesian Coordinates

When a 3D point is projected onto a normalized image plane, information about the 3D nature of the point is lost. That is, given coordinates on the normalized image plane, these

coordinates represent the projection of any of the 3D points lying along the ray passing through it and the camera center. In other words, any 3D point lying along that ray will project to the same point on the normalized image plane. However, if, in addition to the projection coordinates, the range from the camera center were known, then the 3D point could be represented uniquely. Thus, projection-range coordinates can be equivalent to Cartesian coordinates (except for some singularities), given that all points of interest lie in a hemisphere.

There are singularities which occur in the projection-range coordinate representation that do not occur in Cartesian coordinates. These singularities occur when converting from Cartesian coordinates to projection-range coordinates. If a Cartesian point's z-value is zero, then the projection of that point onto the image plane is undefined. If the Cartesian point's x- or y-value is at $\pm\infty$, the projection is also undefined. In addition, an ambiguity occurs when the points span different hemispheres. A projection point can represent points both in front of and behind the camera center. Given that points are not at zero or infinite, and all lie in a common hemisphere, the representation is unique and one-to-one.

The equation for converting a Cartesian coordinate χ_{ij} to projection-range coordinates $\mathbf{X}_{ij}$ is given by

$$\mathbf{X}_{ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ \lambda_{ij} \end{bmatrix} = \begin{bmatrix} \dfrac{\chi_{ij_x}}{\chi_{ij_z}} \\ \dfrac{\chi_{ij_y}}{\chi_{ij_z}} \\ \sqrt{\chi_{ij_x}^2 + \chi_{ij_y}^2 + \chi_{ij_z}^2} \end{bmatrix}. \tag{2.7}$$

To convert from projection-range coordinates to Cartesian coordinates,

$$\chi_{ij} = \begin{bmatrix} \chi_{ij_x} \\ \chi_{ij_y} \\ \chi_{ij_z} \end{bmatrix} = \begin{bmatrix} x_{ij} \dfrac{\lambda_{ij}}{\sqrt{x_{ij}^2 + y_{ij}^2 + 1}} \\ y_{ij} \dfrac{\lambda_{ij}}{\sqrt{x_{ij}^2 + y_{ij}^2 + 1}} \\ \dfrac{\lambda_{ij}}{\sqrt{x_{ij}^2 + y_{ij}^2 + 1}} \end{bmatrix}. \tag{2.8}$$

Both (2.7) and (2.8) apply unambiguously for points lying in the hemisphere directly in front of the camera. Both coordinate systems are relative to the camera center, with orientation aligned with the look direction and the image axes. The concept of projection-range coordinates is shown in Fig 2.9.

2.4.3 Normalized Image Plane and Column-Row Coordinates

Because the EO image is a calibrated image, the x-y coordinates on the normalized image plane can be mapped to the column-row pixel coordinates of a digital image. This is accomplished by using the intrinsic camera calibration matrix,

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

where α_x and α_y are "focal distances" in the x- and y-directions, respectively, s represents the skew of the x- and y-axes, and (x_0, y_0) represents the principal point (intersection of the look direction with the digital image). It is often assumed the skew is zero. To make these concepts more concrete, α_x is a scale factor relating "distance" in the normalized image plane to pixel "distance" in the associate digital image. The same goes for α_y. The principal point can be considered the pixel coordinates of the "center" of the digital image. "Center" in this context does not mean the middle pixel of the digital image, but rather the pixel coordinates which represent the origin of the normalized image plane; i.e., the intersection of the principal ray and the normalized image plane.

For the second-generation texel camera, these quantities are determined to be

$$K = \begin{bmatrix} 1361.25 & 0 & 523.49 \\ 0 & -1361.25 & 538.10 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.10}$$

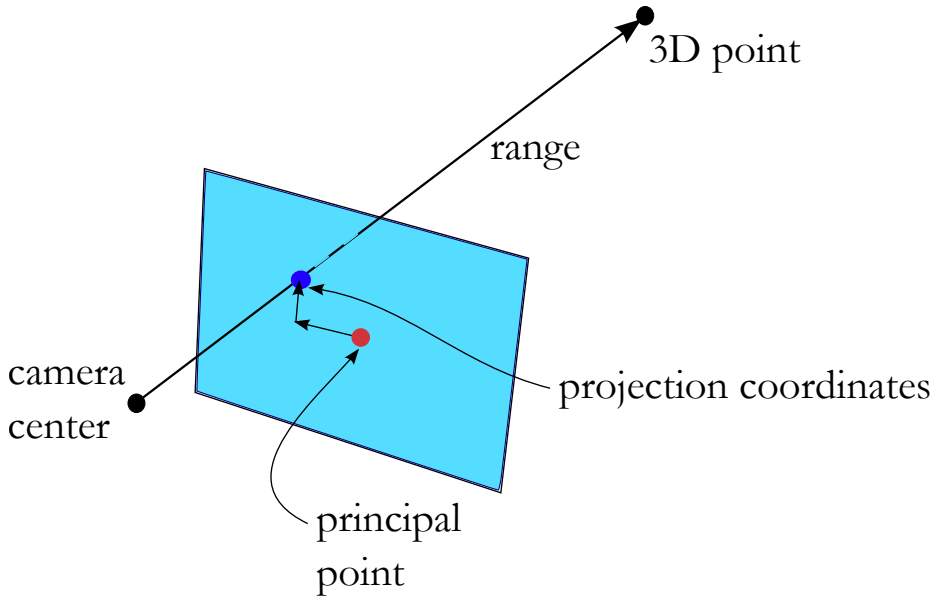


Fig. 2.9: Projection-range coordinate system.

The negative on α_y is due to the fact that the row-coordinate increases in the downward direction while the analogous y-coordinate increases in the opposite direction. The principal point for a given digital image can change when rows and columns are "trimmed" from the image. If the EO image is "zoomed" then the focal distances change, although this is usually not a concern.

In a digital system, pixels are indexed in an array using integers. These digitized pixels are sampled representations of the continuous column-row coordinate system. A location on an image can be a non-integer pixel location. If a pixel value needs to be interpolated between integer locations, there are established methods of performing interpolation [39].

A pixel $\mathbf{p} = [c, r, 1]^T$ can be mapped to normalized image plane coordinates $\mathbf{n} = [x, y, 1]^T$ by applying the inverse camera calibration matrix $\mathbf{n} = K^{-1}\mathbf{p}$. Similarly, normalized image plane coordinates can be mapped to a pixel value by $\mathbf{p} = K\mathbf{n}$.

Thus the normalized image plane coordinates and the column-row coordinates are equivalent representations of a given projection point for a calibrated camera. A graphical representation of their relationship is shown in Fig. 2.10. This relationship enables, for

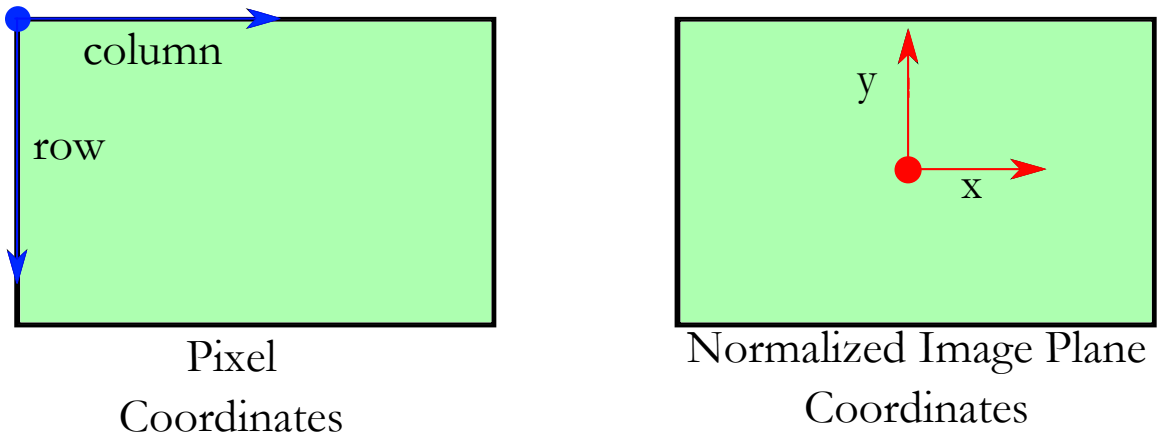


Fig. 2.10: A comparison of image coordinate systems. The left image shows the column-row coordinate system, with the origin as the upper-left pixel. The right image shows the normalized image plane coordinate system, with the origin being the principal point.

example, given a 3D point in space, to find the pixel location of its projection onto the image.

2.5 Homography

A homography describes how the pixel coordinates of a given image are related to the pixel coordinates in another image of the same scene. In other terms, if there are two images of the same scene, a point in the first image can be mapped to the same point in the second image using the homography relationship; i.e., it is a 2D coordinate system transformation. However, the scene (3D world which the image represents) needs to have certain properties for the homography to apply. A homography can be formed between two images when there is no parallax in the images. This restricts the scene to be a plane. The idea of using a homography to map corresponding points from one image to another is shown in Fig. 2.11.

The planar scene assumption is important as it allows the relationship to be linear. Most scenes are not planar, but can, however, be approximated by a homography relationship. The discussion in this section will assume a planar scene to describe the theory. The only exception to the planar-scene requirement for the homography is when images are taken when the only motion the cameras have undergone is rotation about the camera

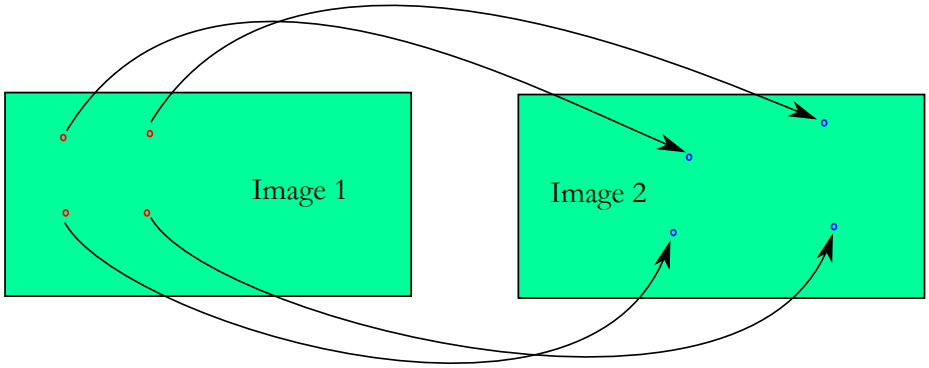


Fig. 2.11: Mapping corresponding points using a homography. The red points in Image 1 are mapped to the blue points in Image 2 using a homography.

center. No parallax is introduced in this case, and a homography holds. This is the idea behind panoramic photos.

Pixel coordinates are often given by the column-row indices (a 2D coordinate system) of the digitized image. A homography is simply a 2D coordinate system transformation applied to digital images. A pixel coordinate, though, needs to be expressed as a homogeneous quantity, in which the third element is equal to a scale factor (usually one); for instance, the 2D pixel $\mathbf{p}_1 = [c_1, r_1, 1]^T$. A homography, then, is a 3×3 matrix in the case of a 2D coordinate system.

2.5.1 Types of Homography

There are several types of homographies, each with a different number of degrees of freedom. When a homography is applied to an image, the image becomes warped. The magnitude of warping is defined by how well lengths and angles are preserved when the homography is applied. In each type of homography listed below, a square is warped per the homography parameters and the resulting shape is described. A homography is unique only to a scale factor, which is apparent to anyone familiar with homogeneous coordinates. Thus, in general, the relationship $\mathbf{p}_2 = H\mathbf{p}_1$ does not hold and should more correctly be written as $w\mathbf{p}_2 = H\mathbf{p}_1$ where w is a scale factor. Alternatively, this can be written as $\mathbf{p}_2 \cong H\mathbf{p}_1$. For certain matrix structures, $w = 1$ as explained below.

The simplest type of homography (apart from the identity transform) is a translation

$$\mathbf{p}_2 = \begin{bmatrix} c_2 \\ r_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_c \\ 0 & 1 & t_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ r_1 \\ 1 \end{bmatrix} = H\mathbf{p}_1 \tag{2.11}$$

where c_m and r_m represent the column-row coordinates in the image m. A square remains a square in this transformation, although it is translated from its original location by (t_c, t_r). The scale factor $w = 1$ is due to the bottom row of H being $[0, 0, 1]$.

The translation homography can be combined with a rigid rotation to form the Euclidean homography

$$H = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & t_c \\ \sin(\theta) & \cos(\theta) & t_r \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.12}$$

A square remains a square, but is rotated and translated in the Euclidean homography.

With increasing complexity, a similarity transform can be introduced

$$H = \begin{bmatrix} s\cos(\theta) & -s\sin(\theta) & t_c \\ s\sin(\theta) & s\cos(\theta) & t_r \\ 0 & 0 & 1 \end{bmatrix} \tag{2.13}$$

which means size can change, depending on the value of s. A square remains a square, but a different size, in addition to any rotations and translations.

An affine transform is given by

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

where each matrix elements loses its individual meaning. A square becomes a parallelogram, in addition to any scaling, rotations, and translations.

A scale-invariant projective transform is given by

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \tag{2.15}$$

where the resulting 3×1 vector must be normalized (by the scale factor) such that its third element is equal to 1. The scale factor is related to the non-zero elements in the bottom row and the point $\mathbf{p}_1$ by $w = (h_{31}c_1 + h_{32}r_1 + 1)$, and is $w \neq 1$ in general. Most of the homography relationships in this document will be referring to the scale-invariant projective transform. In this case, a square becomes a quadrilateral with scaling, rotation, and translation.

The idea of a homography can be extended to 3D points (in homogeneous coordinates) and H becomes a 4×4 matrix. For coordinate system transformations described in Section 2.3.3, these are a 3D form of a Euclidean homography, where angles and lengths are preserved.

2.5.2 Finding a Homography using Matching Points

A homography between two images can be determined by choosing many matching points and finding a least-squares solution using the singular value decomposition (SVD). This derivation will use the scale-invariant projective transform, but can be applied to the other transforms.

Transforming a point $\mathbf{p}_1$ using a projective transform is given by

$$w\mathbf{p}_2 = w \begin{bmatrix} c_2 \\ r_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ r_1 \\ 1 \end{bmatrix} = H\mathbf{p}_1. \tag{2.16}$$

To find the elements of the matrix H, write the equations for c_2 and r_2 explicitly

$$c_2 = \frac{1}{w}(h_{11}c_1 + h_{12}r_1 + h_{13}), \tag{2.17}$$

$$r_2 = \frac{1}{w}(h_{21}c_1 + h_{22}r_1 + h_{23}), \text{ and,} \tag{2.18}$$

$$w = (h_{31}c_1 + h_{32}r_1 + 1). \tag{2.19}$$

Multiplying each side of (2.17) and (2.18) by w, and substituting (2.19) then gathering the h_{ij} terms

$$-h_{11}c_1 - h_{12}r_1 - h_{13} + h_{31}c_1c_2 + h_{32}c_2r_1 + c_2 = 0, \text{ and} \tag{2.20}$$

$$-h_{21}c_1 - h_{22}r_1 - h_{23} + h_{31}c_1r_2 + h_{32}r_2r_1 + r_2 = 0 \tag{2.21}$$

which can be written in matrix form as

$$\begin{bmatrix} -c_1 & -r_1 & -1 & 0 & 0 & 0 & c_1c_2 & c_2r_1 & c_2 \\ 0 & 0 & 0 & -c_1 & -r_1 & -1 & c_1r_2 & r_2r_1 & r_2 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.22}$$

or

$$A\mathbf{h} = \mathbf{0}. \tag{2.23}$$

Each pair of matching points contributes two rows to A. Since there are eight unknowns in the scale-invariant projective transform, a minimum of four pairs of points are needed to find a solution for $\mathbf{h}$. The least-squares solution is found using the SVD of the matrix A. The solution vector $\mathbf{h}$ is a scalar multiple of the right singular vector corresponding to the

smallest unique singular value. Because scale can easily be dealt with by normalizing the last element to one, the SVD gives good results. It is wise to pre-condition the data before forming the matrix A [40]. Once finding the matrix elements, they can be scaled to any desired degree to allow for any scaling present in the homography.

Because there may be noise or inaccuracies in the column-row coordinates of the matching points, it is surmised that if many more points are selected than the minimum number needed for the calculation, the solution is better. However, solving $A\mathbf{h} = \mathbf{0}$ does not account for the presence of any outlier points in the data set. These outliers can be filtered out using the Random Sampling Consensus algorithm, and finding matching points is given in Sections 2.6 and 2.7.

2.5.3 Random Sampling Consensus

Random Sampling Consensus (RANSAC) is a technique used to find the best solution for a given model in the presence of outliers [41]. Given a large data set to model, RANSAC iteratively takes small subsets of the data, finds the model which best fits the subset, and determines the number of inliers of the whole set based on an inlier threshold. It stores the best model, in terms of number of inliers, for the data and returns its value.

In the application of finding a homography using matching points, RANSAC randomly selects a small number of matching points, then uses the SVD to find the associated model for these randomly selected data. Then it tests this model on *all* the points in the data set. If the cost function between a point using the model and the actual value of the point is less than a threshold, it is considered an inlier. In this case, Euclidean distance is used; in other cases Sampson distance can be used. Once the number of iterations is exhausted or a good model is found, the homography with the most number of inliers (or the least error in case of a tie) is considered the best homography for the data.

2.6 Harris Feature Points

A digital image often contains information much greater than its individual pixel values. An image can convey essence. This is evident when a human focuses his or her eyes on

the person in a portrait instead of the blurry background. The *collection* of pixels convey meaning. Digital processing of images, on the other hand, does not consider essence or meaning as humans do. However, mathematically different pixels or areas of an image can have greater value than other pixels or areas. "Value" is a subjective term and, for a researcher, depends on the specific application. The desired goal is to find unique points in images which can be used to find a homography relationship. These unique points are feature points.

There are many different types of feature points, and they can represent edges, corners, blobs, and, more generally, interest points. Each has its strengths and weaknesses. Feature point-finding is a strong research area. The discussion here will focus on a simple, well-established feature point finder, the Harris Corner Detector.

The Harris Corner Detector [42], developed in the 1980s, finds corners of objects in images. "Corner" is a subjective term, but the Harris Corner Detector attempts to describe a corner as a pixel that represents large gradients in both directions. Because of aliasing, resolution, blurring, rotation, and other image non-idealities, a given pixel identified as a corner in one image may not be identified as a corner in another image of the same scene, although the images may look similar. If these non-idealities are small, then corners will be found in similar places of images of the same scene. The Harris Corner Detector needs to be tuned for particular applications and types of imagery.

The Harris Corner Detector algorithm is given in Algorithm 1.

Harris feature points will often occur at common points in two images of the same scene taken from different perspectives, although rotation, lighting, and other non-idealities can cause this to fail. However, if the perspectives are close, then there is a high likelihood that feature points will represent common points between the images. Given that two feature points match, they can be used to find a homography relationship between two images. The criteria for matching is discussed in 2.7.

Algorithm 1 Harris Corner Detector Algorithm

1. Compute the image gradients in the row and column directions, D_r and D_c.
2. Multiply point-by-point to find image planes $S_{rr} = D_r \times D_r$, $S_{cc} = D_c \times D_c$, and $S_{cr} = D_r \times D_c$.
3. Gaussian filter S_{rr}, S_{cc}, and S_{cr} to remove high-frequencies.
4. Compute detector response point-by-point $R = (S_{cc}S_{rr} - S_{cr}S_{cr}) - k(S_{cc} + S_{rr})^2$ where k is on the range $(0.04 - 0.06)$. An alternative detector response is given by Noble [43] in her thesis: $R = \dfrac{S_{cc}S_{rr} - S_{cr}S_{cr}}{S_{cc} + S_{rr}}$. Note this response does not depend on an arbitrarily-picked or empirically-determined constant k.
5. Find each peak in the detector response above a threshold in a given area. Each of these peaks is considered a corner or Harris feature point.

2.7 Normalized Cross-Correlation

Normalized cross-correlation (NCC) determines the similarity between two image patches so patches can be matched to one another. It is similar to 1D correlation (time-reversed convolution), but generalized to 2D signals (imagery). Like in 1D, the fast Fourier transform (FFT) can be used for computation of 2D correlation, although there are some caveats. The image patches of interest do not need to be square, but they need to be of the same dimensions. It is normalized in the sense that it returns a value on the interval $[-1, 1]$, such that different pairs of patches can be compared on the same scale.

For two patches centered about by a given pixel $\mathbf{p}_1 = (c_1, r_1)$ in the first image and $\mathbf{p}_2 = (c_2, r_2)$ in the second image, the NCC score is given by

$$\gamma(\mathbf{p}_1, \mathbf{p}_2) = \frac{\sum_{r=-\frac{N}{2}}^{\frac{N}{2}} \sum_{c=-\frac{M}{2}}^{\frac{M}{2}} (\mathrm{Im}_1(c_1 + c, r_1 + r) - \mu_1)(\mathrm{Im}_2(c_2 + c, r_2 + r) - \mu_2)}{\sqrt{\left(\sum_{r=-\frac{N}{2}}^{\frac{N}{2}} \sum_{c=-\frac{M}{2}}^{\frac{M}{2}} (\mathrm{Im}_1(c_1 + c, r_1 + r) - \mu_1)^2\right)\left(\sum_{r=-\frac{N}{2}}^{\frac{N}{2}} \sum_{c=-\frac{M}{2}}^{\frac{M}{2}} (\mathrm{Im}_2(c_2 + c, r_2 + r) - \mu_2)^2\right)}} \tag{2.24}$$

where $\mathrm{Im}_j(\cdot, \cdot)$ represents image j, and μ_j represents the mean pixel value of the patch j. Typically, $N = M$ making a square patch.

If NCC is done on a color image, each of the color planes is done individually, then the scores are averaged or summed. Normalized cross-correlation is not rotation-invariant; i.e., when comparing the same image patch, it does not indicate a match when the patches are rotated from one another about the center pixel. This can cause problems when trying to find a homography between images with significant rotation. In addition, significant lighting changes within the patches can cause NCC to give bad results. Perspective differences and bland image areas can also cause the NCC to give low scores.

When the NCC score is above a threshold, the points can be considered putative correspondences. In other words, the points probably represent the same point in the scene because their NCC score is high. One can hold the patch Im_1 stationary while moving the patch Im_2 around in a small area (i.e. changing its center coordinates (c_2, r_2)), searching for the best NCC score. This is shown in Fig. 2.12, with the top image as Im_1 and the bottom image as Im_2. The best match for Im_1 will be the location of Im_2 that gives the best NCC score. The centers of each patch can be considered putative correspondences.

Improvements to NCC may include those listed by Giachetti [44]. Because the end goal is to match image patches, perhaps a more computationally-effective method, especially

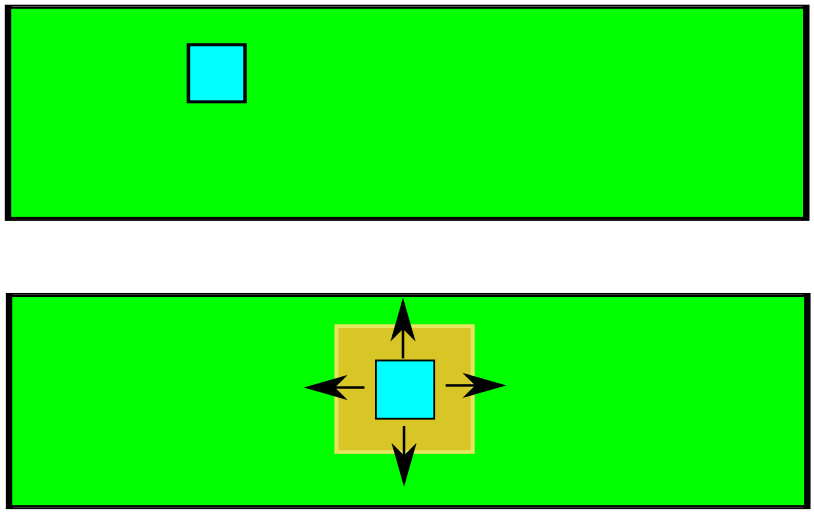


Fig. 2.12: Comparing two patches in two images using NCC. The patches (blue) in each image are compared using NCC. The patch in the top image remains stationary while the patch in the bottom image is moved around in a small area (tan) to find the best NCC score.

when searching a large area for a match, is to use a method based on the 2D FFT, such as Partial-Aliasing Correlation Filters [45].

2.8 Fundamental Matrix and Epipolar Geometry

The fundamental matrix defines the relationship between a pair of cameras in 3D space. It relates the projection of a 3D point in one image to a line of possible projections in another image. This is more general than a homography as it applies for non-planar scenes. However, unlike a homography, the fundamental matrix is used to map a point in one image to a line in the other.

Consider the projection of a 3D point on an image. This projection point is the projection of any 3D point along the ray going from the camera center, through the projection point, and out to infinity. In other words, any 3D point along that ray projects to the same point on the normalized image plane. It is impossible to know anything about the 3D point (except that it lies along the afore-mentioned ray) without additional information.

In order to increase the amount of information present, a second camera is added viewing the same scene from a different location. This camera can see the ray protruding out from the first camera. This ray is projected onto the second camera's image. Thus, a point in the first image maps to a line in the second image. This also applies in the reverse.

A plane can be created by the two camera centers and any point along the out-going ray. This plane is called an epipolar plane. Each camera can see the other camera center as a projection on its image plane. The projections of these camera centers into the other image are called epipoles. If the cameras are simply translated from one another, the epipoles lie at infinity. The epipoles do not necessarily need to lie on the digital image, as a digital image represents only a subset of the normalized image plane. These concepts are shown in Fig. 2.13.

The fundamental matrix is a mathematical relationship between a point projection in one image, the camera locations, and the point's corresponding epipolar line in the other image. The 3×3 fundamental matrix F relates a point $\mathbf{p}_1 = [c_1, r_1, 1]^T$ on the epipolar

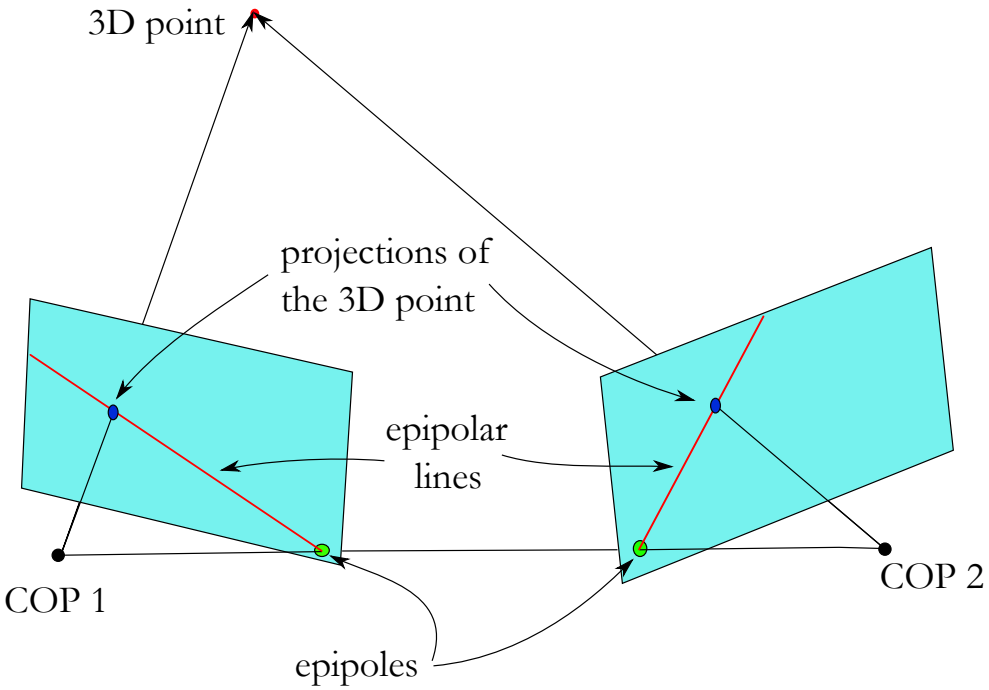


Fig. 2.13: Two-camera epipolar geometry. The red epipolar lines represent the projection of the rays going to a 3D point in space. The projection of the other camera center into each image is an epipole, represented by green dots. An epipolar plane is formed by the camera centers and the 3D point.

line in the first image to a corresponding point $\mathbf{p}_2 = [c_2, r_2, 1]^T$ on the epipolar lines in the second image by the relationship

$$\mathbf{p}_2^T F \mathbf{p}_1 = 0, \tag{2.25}$$

known as the epipolar constraint.

Examining this closely shows that if a point $\mathbf{p}_1$ is known, then the location of $\mathbf{p}_2$ can be constrained to a line. That is, let $\mathbf{k} = F\mathbf{p}_1$ where $\mathbf{k} = [a, b, c]^T$, then the epipolar constraint can be expressed as

$$\begin{bmatrix} c_2 & r_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0 \tag{2.26}$$

which defines a 2D line on the normalized image plane in the second image. A similar equation can be defined when $\mathbf{p}_2$ is known.

The epipoles in each image are the left and right nullspace of the fundamental matrix, and are found using the SVD.

2.8.1 Computing the Fundamental Matrix

Computing the fundamental matrix can be done in two ways: (1) from the relative position and orientation of two calibrated cameras, or (2) from corresponding points. The computation of the fundamental matrix from camera location, attitude, and calibration is simply

$$F = (K_2^{-1})^T [\mathbf{t}]_\times R K_1^{-1} \tag{2.27}$$

where K_1 and K_2 are the camera calibration matrices for the first and second cameras, respectively, R is the relative rotation between the camera centers, and $\mathbf{t}$ is the relative translation between the camera centers, in the coordinate system the first camera. The notation $[\mathbf{t}]_\times$ denotes a skew-symmetric matrix formed from a vector $\mathbf{t} = [t_1, t_2, t_3]^T$ in a

cross-product formulation, given by

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}. \tag{2.28}$$

The matrix $E = [\mathbf{t}]_\times R$ is called the essential matrix, which contains the rotation and translation between two cameras. The essential matrix has a comparable property in which

$$\mathbf{n}_2^T E \mathbf{n}_1 = 0 \tag{2.29}$$

where $\mathbf{n}_i$ are expressed in normalized image coordinates. Both F and E are rank two matrices.

The method using corresponding points, commonly called the Eight-Point Algorithm, was first introduced by Luong et al. [46] and later improved by Hartley [40]. This algorithm will not be presented here, but has a formulation similar to that of finding a homography from corresponding points. It minimizes the Sampson distance, which is the distance between a point and its corresponding point's epipolar line.

2.8.2 Recovering Rotation and Translation from the Fundamental Matrix

Because the fundamental matrix is formed from both the intrinsic camera calibration matrices and relative positioning of the cameras, it makes sense that the relative position and orientation of the cameras can be recovered given the calibration matrices are known. The proof will not be given here, but the process is shown below and is found in Hartley and Zisserman [47, p. 258].

The fundamental matrix can be decomposed into the essential matrix by

$$E = (K_2)^T F K_1. \tag{2.30}$$

The essential matrix has three singular values. Two singular values are equal and the third is zero. The SVD of the essential matrix E can be written as $E = U\text{diag}(\alpha, \alpha, 0)V^T$. Because two of the singular values are equal, the SVD is not unique. Let

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.31}$$

and the left SVD matrix U be decomposed into its columns

$$U = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}. \tag{2.32}$$

The relative rotation matrix between the two cameras is given by either

$$\hat{R}_1 = UWV^T \tag{2.33}$$

or

$$\hat{R}_2 = UW^TV^T. \tag{2.34}$$

This ambiguity is due to the properties of the essential matrix having two equal singular values. The direction of the translation is given as

$$\hat{\mathbf{t}} = \beta\mathbf{u}_3, \tag{2.35}$$

however, the magnitude of the direction β is not known. This is due to the fact that the essential matrix is unique only to scale. Thus, a priori knowledge of the rotation and translation must be known to reduce the ambiguity for R and to find an estimate of the magnitude β. An alternative method to recovering the rotation and translation from the essential matrix is given by Horn [48], but still has these ambiguities.

Reducing Ambiguity for Recovering Rotation and Translation

If the fundamental (and hence, essential) matrix is found using corresponding points from calibrated cameras, and there is some coarse knowledge of the relative location $\tilde{\mathbf{t}}$ and orientation $\tilde{R}$ of the two cameras, then the rotation ambiguity can be reduced and the translation magnitude can be determined.

After following the afore-mentioned process of finding the recovered matrices $\hat{R}_1$ and $\hat{R}_2$, it logically follows that if $\tilde{R}$ is a good estimate, then the recovered matrix that represents an orientation in approximately the same direction as $\tilde{R}$ is the better choice. This can be done by taking a vector $\mathbf{x} = [x, y, z]^T$ and rotating it by $\tilde{R}$, $\hat{R}_1$, and $\hat{R}_2$. This results in vectors $\tilde{\mathbf{x}}$, $\mathbf{x}_1$, and $\mathbf{x}_2$, respectively. The dot products $d_1 = \tilde{\mathbf{x}} \cdot \mathbf{x}_1$ and $d_2 = \tilde{\mathbf{x}} \cdot \mathbf{x}_2$ are computed. If $d_1 \geq d_2$ then $R = \hat{R}_1$ is chosen, and $R = \hat{R}_2$ is chosen if otherwise.

Finding the magnitude β makes more assumptions. If the initial translation measurement $\tilde{\mathbf{t}}$ is assumed to be made with isotropic noise, then the guess $\tilde{\mathbf{t}}$ can be projected onto $\hat{\mathbf{t}}$ to find the optimal vector length, which eliminates the need to find the numerical value for β. This concept is shown in Fig. 2.14. The recovered translation $\mathbf{t}$ is given by

$$\mathbf{t} = \frac{\tilde{\mathbf{t}} \cdot \hat{\mathbf{t}}}{||\hat{\mathbf{t}}||^2} \hat{\mathbf{t}}. \tag{2.36}$$

Thus, the recovered rotation and translation is $[R|\mathbf{t}] = [\hat{R}_i|\frac{\tilde{\mathbf{t}} \cdot \hat{\mathbf{t}}}{||\hat{\mathbf{t}}||^2}\hat{\mathbf{t}}]$ where i is determined above.

Alternate Method to Recover Translation

This method eliminates the need to find the SVD of the essential matrix. The essential matrix can be decomposed into an orthonormal rotation matrix and a skew-symetric matrix, $E = [\mathbf{t}]_\times R$. If E is found using corresponding points, and R is known to some degree by, for example, a physical measurement, then an estimate of the translation can be determined by finding the matrix product $\hat{T} = ER^T$. The resulting matrix $\hat{T}$ is not guaranteed to be skew-symmetric (in the presence of noise and errors), but can be projected onto a skew-symetric

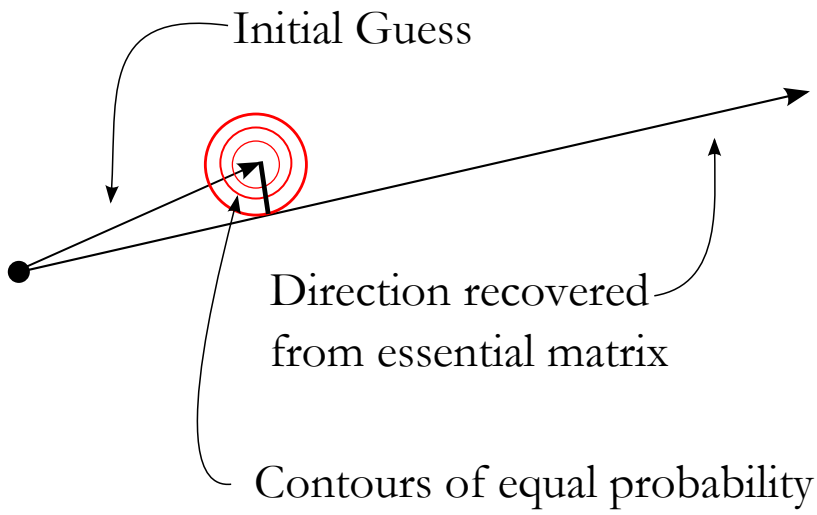


Fig. 2.14: Finding the best translation given isotropic noise on the initial guess. The red concentric circles represent isotropic measurement noise.

matrix by

$$[\hat{\mathbf{t}}]_\times = \begin{bmatrix} 0 & -\frac{\hat{T}_{21}-\hat{T}_{12}}{2} & \frac{\hat{T}_{13}-\hat{T}_{31}}{2} \\ \frac{\hat{T}_{21}-\hat{T}_{12}}{2} & 0 & -\frac{\hat{T}_{32}-\hat{T}_{23}}{2} \\ -\frac{\hat{T}_{13}-\hat{T}_{31}}{2} & \frac{\hat{T}_{32}-\hat{T}_{23}}{2} & 0 \end{bmatrix} \tag{2.37}$$

where $\hat{T}_{ij}$ is an element of the matrix $\hat{T}$. The vector $\hat{\mathbf{t}}$ can be written as

$$\hat{\mathbf{t}} = \begin{bmatrix} \frac{\hat{T}_{32}-\hat{T}_{23}}{2} \\ \frac{\hat{T}_{13}-\hat{T}_{31}}{2} \\ \frac{\hat{T}_{21}-\hat{T}_{12}}{2} \end{bmatrix}. \tag{2.38}$$

Because both the fundamental and essential matrices are unique up to a scale factor, the vector $\hat{\mathbf{t}}$ is merely the direction of the translation, and additional means need to be employed to determine the magnitude, such as that in (2.36).

2.8.3 Applications of the Fundamental Matrix

The fundamental matrix is a useful tool in image processing because it applies to scenes that are not planar, loosening the restriction set by the homography. The fundamental

matrix can be known to some degree of accuracy, either through knowledge of the camera or from knowledge of corresponding points.

If the fundamental matrix is known for calibrated cameras, then the essential matrix can be determined. Given the essential matrix, relative positions of cameras can be determined. This allows for 3D reconstruction, and is the basis for many photogrammetric techniques.

2.9 Conclusion

The texel camera concept allows for 3D information and EO imagery to be calibrated upon capture. Because the EO image is calibrated, a pixel in a digital image can be mapped to the normalized image plane through a previously known mapping. In addition, 3D information can be preserved during a projection onto the normalized image plane by retaining its range information. Relationships between the 2D projections and 3D world can be developed. Image processing techniques allow for relationships between images to be found and exploited, giving greater information than was available from individual images.

Chapter 3

Triangulation of Texel Swaths

3.1 Swath Registration and 3D Reconstruction Problem

Given two texel images, each containing both a point cloud and an electro-optical image which are calibrated to one another, both 2D- and 3D-image registration techniques can be used to combine the two texel images into a meaningful, detailed representation of a 3D scene. Methods are being developed for registering 3D data with 2D imagery [49]. Past methods using texel images have used an entire array in the registration process [50] as well as including a coarse position and attitude estimate [51].

For most instruments, however, a large point cloud representing the scene cannot be captured all at once in an aerial vehicle. Only a subset, a small number of ladar strips, can be gathered in real-time from moving, small UAVs. In addition, position and attitude instruments on a typical small UAV are not accurate enough to make ladar measurements useful on their own. Although the ladar measurements may be accurate to within a few centimeters, the reference from which the measurements are taken may not be accurate to that degree.

Thus, in order to make the measured laser data useful, the location of the UAV must be localized to near the accuracy of the ladar measurements. This is a problem with images from small UAVs in general, as Kung et al observe [52]. This problem is mitigated by using multiple texel swaths to triangulate their relative positions. This, in essence, is a camera pose recovery problem using photogrammetry, but with additional 3D information.

Each texel swath contains several pieces of information: an EO image, measured ladar points calibrated to the EO image, and the approximate location and attitude at which the

points were measured. The texel camera also has calibration parameters that allow a point in 3D space to be mapped to a row-column location in the EO image.

3.2 Triangulation of 3D Points

Given two texel swaths, j and k, examining the same scene from different perspectives, the 3D points in j can be projected into both the j^{th} and k^{th} normalized image planes. Also, the 3D points in k can be projected into both the k^{th} and j^{th} normalized image planes. In other words, each swath can see its own 3D points and the 3D points of its neighbor. Additionally, the images of a given 3D point in each texel swath should look similar to one another, with any differences due to the swath perspectives. Image processing techniques can be used to match these images. Adjacent texel swaths share a lot of information; they look at the same scene, they can see many (if not all) of each other's 3D points, and their EO imagery is similar.

Image processing techniques can be used to find the projection matches for a 3D point i in several texel swaths. That is, these techniques find corresponding image patches in several swaths that represent this 3D point. In addition, the measured location and attitude of each camera can be used to determine the 3D-to-2D projection of the point i into several texel swaths. These two methods of finding projections of a single point into an image will usually give different results, and the difference between the projections can be considered error. An example of this is shown in Fig. 3.1.

These two pieces of knowledge can be used to formulate a cost function for the system. This cost function is described in the remainder of this chapter, and is formulated in a system where there are many 3D points and many cameras. A mathematical description of this cost function follows.

Each texel swath j acquires n_j ladar points, a set denoted as I_j. By mapping all the ladar points in M swaths into a common coordinate system $\mathcal{O}$ using the coarse location/attitude measurements, the entire set of 3D points can be indexed using i, and each point is represented as a 3D vector of Cartesian coordinates, $\mathbf{b}_i = [b_{i_x}, b_{i_y}, b_{i_z}]$. The total number of 3D points in the system is $N = \Sigma_{j=0}^{M-1} n_j$. A ladar point $\mathbf{b}_i$ belongs to the texel

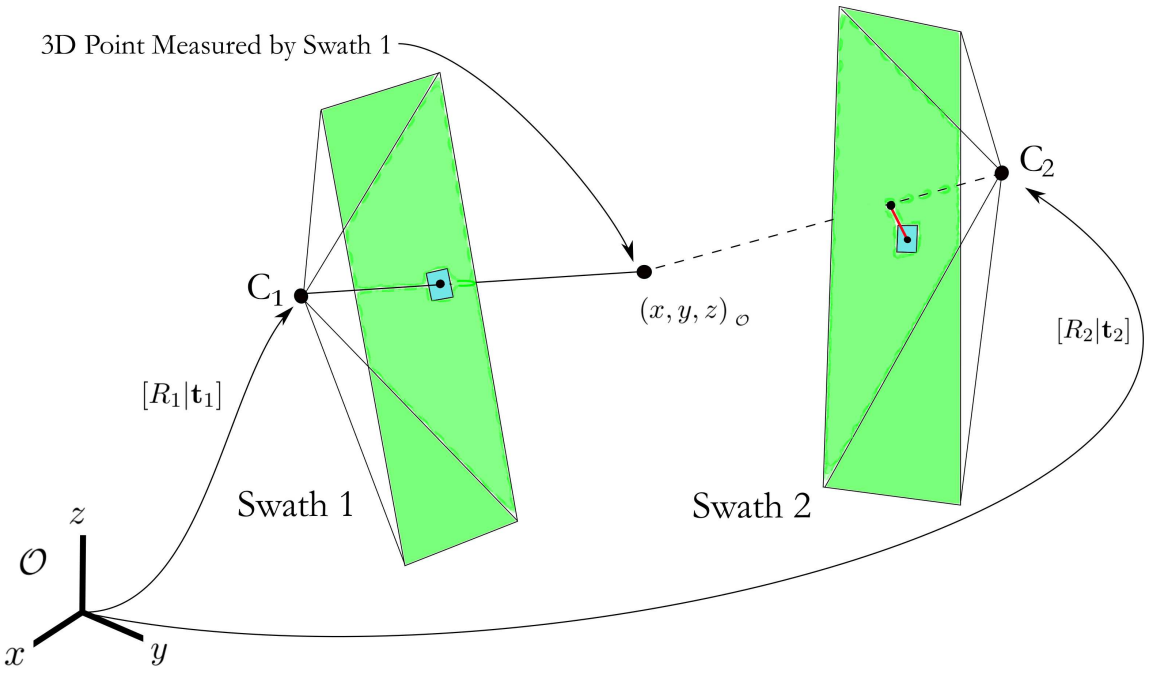


Fig. 3.1: Projection of a 3D point into image swaths. C_1 and C_2 represent the camera centers. The 3D point is in the set I_1 but is represented in the world coordinate system as $(x, y, z)_{\mathcal{O}}$ in this illustration. The blue patches in each swath represent matching image patches. The red vector in Swath 2 is the projection error.

swath j if $i \in I_j$. Calibrated 2D projection points are defined as the projection of points $\mathbf{b}_i$ into the normalized image plane of texel swath j when $i \in I_j$. These calibrated 2D projections of a 3D point are found during the camera calibration and do not change from image-to-image for a given depth pixel. Because the EO image is calibrated, the normalized projections can be mapped to pixel coordinates using (2.9).

Each 3D point i can be rotated and projected into each image plane j by using the methods described in Sections 2.3 and 2.4. The 3D rotation is given by

$$\hat{\boldsymbol{\chi}}_{ij} = \begin{bmatrix} \hat{\chi}_{ij_x} \\ \hat{\chi}_{ij_y} \\ \hat{\chi}_{ij_z} \end{bmatrix} = \begin{bmatrix} 1 - \frac{2(q_{j2}^2+q_{j3}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j1}q_{j2}-q_{j0}q_{j3})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j1}q_{j3}+q_{j0}q_{j2})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} \\ \frac{2(q_{j1}q_{j2}+q_{j0}q_{j3})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & 1 - \frac{2(q_{j1}^2+q_{j3}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j2}q_{j3}-q_{j0}q_{j1})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} \\ \frac{2(q_{j1}q_{j3}-q_{j0}q_{j2})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & \frac{2(q_{j2}q_{j3}+q_{j0}q_{j1})}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} & 1 - \frac{2(q_{j2}^2+q_{j1}^2)}{q_{j0}^2+q_{j1}^2+q_{j2}^2+q_{j3}^2} \end{bmatrix}^T \begin{bmatrix} b_{i_x} - t_{jx} \\ b_{i_y} - t_{jy} \\ b_{i_z} - t_{jz} \end{bmatrix}, \tag{3.1}$$

and the projection-range conversion is given by

$$\hat{\mathbf{X}}_{ij} = \begin{bmatrix} \hat{x}_{ij} \\ \hat{y}_{ij} \\ \hat{\lambda}_{ij} \end{bmatrix} = \begin{bmatrix} \frac{\hat{\chi}_{ij_x}}{\hat{\chi}_{ij_z}} \\ \frac{\hat{\chi}_{ij_y}}{\hat{\chi}_{ij_z}} \\ \sqrt{\hat{\chi}_{ij_x}^2 + \hat{\chi}_{ij_y}^2 + \hat{\chi}_{ij_z}^2} \end{bmatrix}. \tag{3.2}$$

Both (3.1) and (3.2) apply to $i \in I_j$ and $i \notin I_j$. The vector $\hat{\mathbf{X}}_{ij}$ represents the projection-range coordinates of point i in the j^{th} texel swath. These projection-range coordinates depend on both the 3D point $\mathbf{b}_i$ and the camera parameters $[q_{j0}, q_{j1}, q_{j2}, q_{j3}, t_{jx}, t_{jy}, t_{jz}]$, and any errors in any of these quantities can propagate into the projection and range.

The projection error is the Euclidean distance on the normalized image plane from the correct projection to the actual projection. The correct projection of each point i into the j^{th} image plane is given by x_{ij} and y_{ij}. The values for the correct projections are determined by the relationship a swath j has to the point i. When the point $i \in I_j$, the correct projections are given by the calibration points. For the other case $i \notin I_j$, the correct

projections are determined by image processing techniques. The range error is given as the difference between the measured range and the actual range; the correct range is the measured range. The measured ladar range λ_{ij} for $i \in I_j$ is acquired by the texel camera. There is no measured range when $i \notin I_j$.

Thus, depending on the relationship between the point i and the swath j, two formulations for the error in the system are defined.

For $i \in I_j$

The error for these points consists of the distance from calibrated projection point (x_{ij}, y_{ij}) to the actual projection $(\hat{x}_{ij}, \hat{y}_{ij})$ and the difference between the measured range value λ_{ij} and the actual range value $\hat{\lambda}_{ij}$. The squared error for the ij-combination is

$$\epsilon_{ij}^2 = \frac{1}{\sigma_I^2}(x_{ij} - \hat{x}_{ij})^2 + \frac{1}{\sigma_I^2}(y_{ij} - \hat{y}_{ij})^2 + \frac{1}{\sigma_\lambda^2}(\lambda_{ij} - \hat{\lambda}_{ij})^2 \text{ for } i \in I_j \tag{3.3}$$

where σ_I^2 and σ_λ^2 are the variances of the calibration and range measurements, respectively.

For $i \notin I_j$

The error for these points consists of the distance from the projection point found using image processing techniques (x_{ij}, y_{ij}) to the actual projection $(\hat{x}_{ij}, \hat{y}_{ij})$. Because there is no range measurement there is no error contribution from the range values. The squared error for the ij-combination is

$$\epsilon_{ij}^2 = \frac{1}{\sigma_{\bar{I}}^2}(x_{ij} - \hat{x}_{ij})^2 + \frac{1}{\sigma_{\bar{I}}^2}(y_{ij} - \hat{y}_{ij})^2 \text{ for } i \notin I_j \tag{3.4}$$

where $\sigma_{\bar{I}}^2$ is the variance of the projection points determined by image processing techniques.

These sources of error in the system are shown in Fig. 3.2.

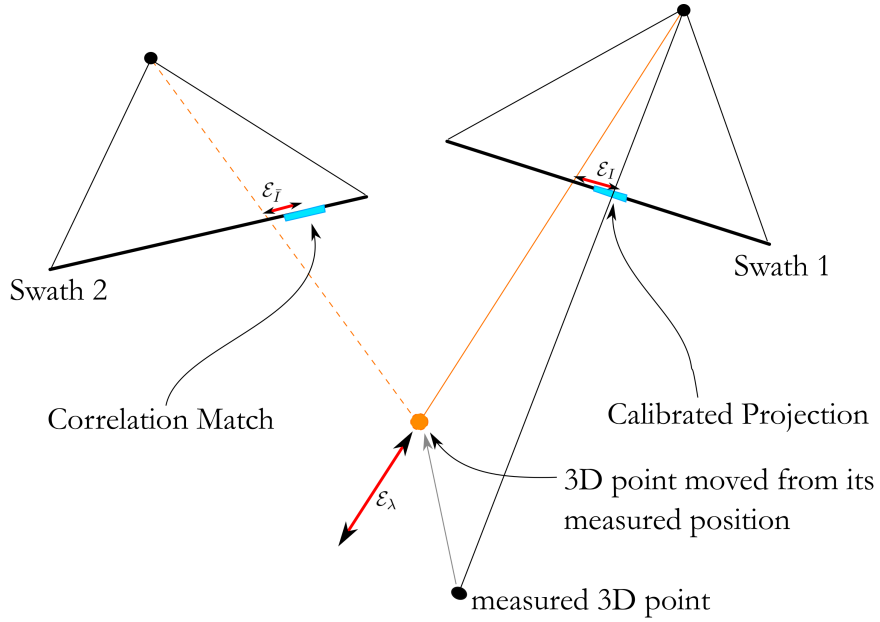


Fig. 3.2: Sources of error in the system. The red vectors indicate possible sources of error: $\mathcal{E}_I$ is the projection distance from the calibrated projection, $\mathcal{E}_\lambda$ is the distance from the measured range, and $\mathcal{E}_{\bar{I}}$ is the projection distance from the matched image patch using correlation. There is no range error present for Swath 2 because it did not measure the 3D point.

These can be combined into a single cost function for the entire system

$$\begin{aligned}\mathcal{E}^2 = &\sum_{j=0}^{M-1}\sum_{i\in I_j}\frac{1}{\sigma_I^2}[(x_{ij}-\hat{x}_{ij})^2+(y_{ij}-\hat{y}_{ij})^2] + \sum_{j=0}^{M-1}\sum_{i\notin I_j}\frac{1}{\sigma_{\bar{I}}^2}[(x_{ij}-\hat{x}_{ij})^2+(y_{ij}-\hat{y}_{ij})^2] \\ &+ \sum_{j=0}^{M-1}\sum_{i\in I_j}\frac{1}{\sigma_\lambda^2}(\lambda_{ij}-\hat{\lambda}_{ij})^2\end{aligned} \tag{3.5}$$

which represents the weighted sum of squared error. If any of the measurements are missing for a given element in each sum, the element is simply omitted from the sum. The formulation of the cost function enables optimization to occur to minimize the sum of squared error in the system. However, before the optimization is described, the image processing techniques used to obtain (x_{ij}, y_{ij}) will be discussed.

3.3 Algorithm for Finding Projection Points

The 3D point $i \in I_j$ has calibrated projection coordinates (x_{ij}, y_{ij}). These calibrated projection coordinates do not change and are considered "ground truth." The 3D point i also projects into a neighboring texel swath k where $k \neq j$. Because the texel swaths j and k are captured from similar perspectives, the imagery surrounding the projections of point i into each swath j and k should look similar. In other words, a small image patch surrounding the projection (x_{ij}, y_{ij}) should look similar to the small image patch surrounding the projection (x_{ik}, y_{ik}). The image patch around (x_{ij}, y_{ij}) can be used as reference because it represents a calibrated projection. All that needs to be done is to search the image k for a small patch that looks like the reference patch in image j. Once this small patch is found, the patch center is appraised to be (x_{ik}, y_{ik}), the correct projection of the 3D point i into the texel swath k. This correct projection is then used in the error calculation (3.5) when $i \notin I_j$.

These correct projection points are found using image processing techniques. The image patches are matched using NCC. However, searching the entire image k to find the small patch with the highest NCC score is unreasonable and requires a lot of computation, in addition to the chance nothing will be found due to image rotation. In order to limit the search area and remove rotation, a homography between the images j and k can be found. Because a homography holds only for planar scenes and the landscapes captured in the texel swaths are not necessarily planar, the homography only serves as an approximation in order to remove rotation and limit the search area.

A homography H_{jk} between the two images j and k is found by finding Harris feature points in each image. Once the Harris features are found, they must be matched to one another to form putatative correspondences. Normalized cross-correlation is used for the matching. Each feature point in j is compared to each feature point in k. This is computationally intensive. However, this process can be expedited by using a knowledge of the fundamental matrix formed from the coarse attitude and location of the texel swaths. If the distance between a feature point in k to the epipolar line in k of a feature in j is greater

than a threshold, then it is assumed that they are not correspondences. After the putative correspondences are found, they are filtered using RANSAC to find the best model while removing the outliers.

Once a homography H_{jk} is found, the image k is registered, warped, and resampled into the same space as image j. This registration enables NCC to work better because the images are in a common image space (there should be little or no relative rotation and distortion).

A small patch in image j around the calibrated projection point (x_{ij}, y_{ij}) is identified and used as the reference patch. The initial search location $\mathbf{n}_{ik_0}$ in image k is determined using

$$\mathbf{n}_{ik_0} = K^{-1} H_{jk} K \mathbf{n}_{ij} \tag{3.6}$$

where $\mathbf{n}_{ij} = [x_{ij}, y_{ij}, 1]^T$, and K is the camera calibration matrix to convert pixel coordinates to normalized coordinates. Because there may be parallax between the images or if H_{jk} is inaccurate, a small area is searched around $\mathbf{n}_{ik_0}$. The pixel location with the highest NCC score above a threshold in this small search area becomes $(\hat{x}_{ik}, \hat{y}_{ik})$. This is the location of the projection of i into the swath k using image processing techniques. These steps are shown in Fig. 3.3.

This process for finding projection points is described in Algorithm 2. The swath registration algorithm depends on the successful completion of this process, as well as the accuracy of the points found therein. Thus, great care must be taken to ensure these projection points found using image processing are reliable. Once these projection points are found, they are converted from pixel coordinates to normalized image coordinates (if not already in normalized image coordinates), and the optimization can begin.

3.3.1 Finding an Initial Homography Estimate from Camera Position and Attitude

The process of finding projection points relies heavily on the fact that a homography can be found between the image swaths j and k. As mentioned in previous sections, Harris

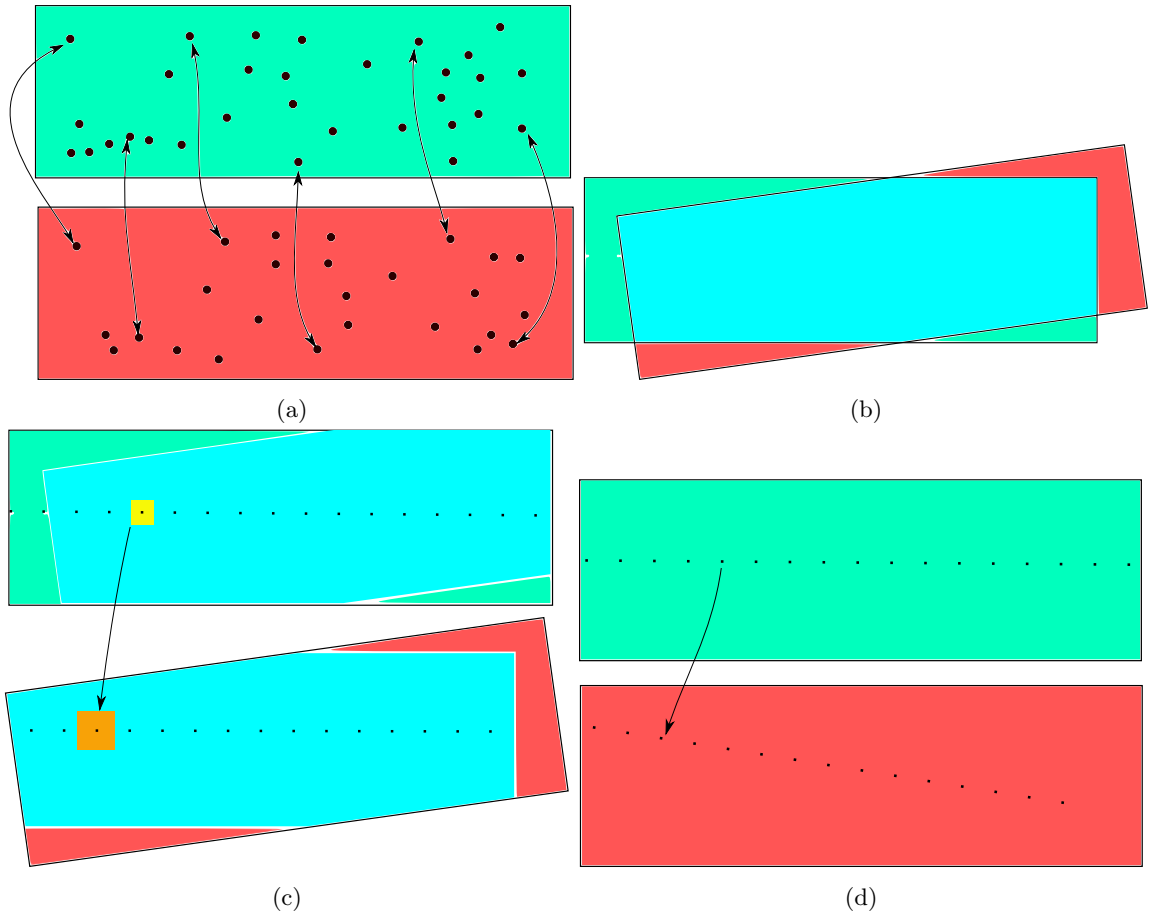


Fig. 3.3: Finding projection points in adjacent swaths. (a) Finding a homography from matching Harris feature points. (b) Warping and resampling images into the same image space. (c) Using correlation to find projection points by matching image patches around calibrated projections. (d) Matching projections found using correlation.

Algorithm 2 Finding Projection Points in Neighboring Swaths using Image Processing

1. Find Harris feature points in adjacent swaths.
2. Find corresponding Harris feature points using NCC.
3. Determine the homography using linear least-squares and RANSAC. This step assumes there is no rotation between the swaths.
4. If Step 3 fails, assume there is significant rotation between the swaths and perform the following.
   (a) Use the position and attitude knowledge from the GPS/IMU project the 3D points from each swath into image of the other swath.
   (b) Using these image points, and the corresponding image points known from the calibration process, compute an initial homography between swaths.
   (c) Using the initial homography, map the second image into the first, and repeat Steps 1-3 with the mapped second image and original first image. This enables the Harris feature finding and NCC to work effectively.
   (d) The final homography used in Step 5 is the cascade of the homographies in Steps 4b and 4c.
5. Determine the projection of each ladar point in the first image, then map each point and surrounding pixels to the other image using the homography found in Step 3 or Step 4.
6. Search an area centered around the expected ladar point location using NCC to find each of the matching locations. Save these locations for optimization.

feature points and NCC often fail when there are rotation or large perspective changes. The perspective changes are not a concern because it is assumed the images are taken one after another from nearly the same perspective. However, there may be rotation between the images which causes NCC to fail while identifying putative correspondences. This can be mitigated by using the position and attitude of the cameras to determine an initial homography estimate. Then the process described above can be used to find H_{jk}.

To find a homography estimate using 3D points and camera knowledge, the 3D points of j are projected into the normalized image plane of j (which are the calibrated projection points). Those same 3D points are moved into the coordinate system of swath k using the estimated position and attitude of each texel swath, and are projected into the normalized image plane of k. Each 3D point in j has a projection in both j and k; i.e. matches in both images. This is shown in Fig. 3.4. A homography estimate $\hat{H}_{jk0}$ can be determined using these points.

Once the homography estimate is found, the image k is resampled such that there is no rotation between it and the image j. This resampled image is referred to as $\hat{k}$. A homography $H_{j\hat{k}}$ can be found between images j and $\hat{k}$ using Harris feature points, NCC, and RANSAC as described earlier in this section.

The final homography can be found by $H_{jk} = H_{j\hat{k}}H_{jk0}$. The projection points can then be found using the methods above.

3.3.2 Finding Projection Points in Non-Adjacent Images

Finding a homography between each pair of images is computationally intensive. However, if pair-wise homographies are found, then the homographies can simply be cascaded and used to find projection points in non-adjacent images. For example, if the homography between swath one and swath two is H_{12} and the homography between swath two and swath three is H_{23}, the homography between swath one and swath three can be approximated by $H_{13} = H_{12}H_{23}$. The homography H_{13} can then be used to find the projections of ladar points for swaths one and three. This can be extended to as many swaths as can be seen

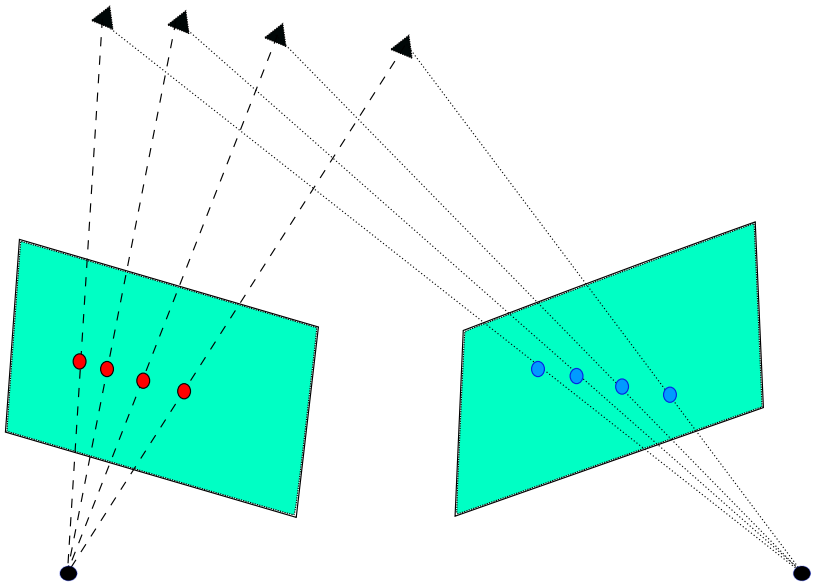


Fig. 3.4: Finding a homography from 3D points and camera information. The red and blue circles represent the projections in each image of the given set of 3D points, and can be used to determine a homography estimate.

by a given swath, although inaccuracies in the homography can accumulate, especially with perspective distortion and parallax.

3.4 Bundle Adjustment Optimization

The optimization is to reduce the error present in the system, as given by (3.5). Because of both the 3D-to-2D projection and the quaternion rotation, this problem is nonlinear, and a nonlinear optimization technique must be used. The term "bundle adjustment" is used to denote that all the data is optimized jointly. One popular method for nonlinear bundle adjustment optimization is the Levenberg-Marquardt Algorithm.

Levenberg Marquardt Algorithm

The Levenberg-Marquardt Algorithm (LMA) is an iterative nonlinear least-squares process of fitting model parameters to a set of data [53]. It is a hybrid of the Gauss-Newton Algorithm and the Gradient Descent method. It starts with an initial guess of the model parameters. The algorithm adjusts the model parameters iteratively to find the minimum error solution.

In each iteration the descent direction (negative gradient) is calculated on an error surface at the current location, which may be multidimensional. Because of the potentially nonlinear nature of the error surface, a local linearization is performed at this location to obtain the gradient. The error is evaluated a small step-size along the descent direction. If the error in the new location is smaller than the current error, the solution moves to the new location and decreases the step-size parameter. Otherwise, it stays in its current location, increases the step-size parameter, recalculates the step-size, and tries again.

After a number of iterations, the error should be close to the desired minimum. However, if the initial guess is too far away from the minimum the algorithm may take a long time to converge or it may get stuck in a local minimum along the way. A criterion for convergence is typically a small change in error reduction.

3.4.1 Sparse Levenberg-Marquardt Algorithm

Hartley and Zisserman [47, p. 611] describe an application of the LMA to image bundle adjustment. In typical LMA algorithms, only the model parameters are adjusted to fit the data. Hartley and Zisserman adjust both the model parameters and the data points to fit one another. This is because the data points are known to have some non-zero variance due to measurement error. This allows both the model and the data to adjusted concurrently, giving the best fit for the system, minimizing error in the model and data. If the model parameters and the data are assumed to have a Gaussian distribution, the minimization of (3.5) is a maximum-likelihood solution. Their application of bundle adjustment takes advantage of the fact that many relationships between the parameters and the data points are nonexistent. This gives sparsity in the problem that can significantly reduce computation.

The system error (3.5) is nonlinear and a large system with sparse relationships, making the Sparse LMA an ideal choice for the optimization. The following mathematical description is the application of the texel swath registration problem to the Sparse LMA.

The model parameters are described using

$$\mathbf{a}_j = [q_{0j}, q_{1j}, q_{2j}, q_{3j}, t_{xj}, t_{yj}, t_{zj}]^T , \tag{3.7}$$

while each data point is described as

$$\mathbf{b}_i = \left[\chi_i \mathcal{O}_x, \chi_i \mathcal{O}_y, \chi_i \mathcal{O}_z\right]^T \tag{3.8}$$

where $\mathbf{a}_j$ represent the attitude and location of the j^{th} texel image, and $\mathbf{b}_i$ represents the Cartesian coordinates of the i^{th} 3D point in the world coordinate system $\mathcal{O}$.

The projection-range coordinates $\hat{\mathbf{X}}_{ij}$ of each point-swath combination is given by (3.2), and each $\hat{\mathbf{X}}_{ij}$ is a function of both $\mathbf{a}_j$ and $\mathbf{b}_i$. In each iteration, the Jacobians

$$A_{ij} = \left[\frac{\partial \hat{\mathbf{X}}_{ij}}{\partial \mathbf{a}_j}\right] \tag{3.9}$$

and

$$B_{ij} = \left[\frac{\partial \hat{\mathbf{X}}_{ij}}{\partial \mathbf{b}_i}\right] \tag{3.10}$$

are calculated for all $i = 1, ..., N$ and $j = 1, ..., M$. These partial derivatives are derived symbolically and hard-coded into the optimization. The matrices A_{ij} and B_{ij} describe the relationship between the i^{th} 3D point and the j^{th} texel swath, for both $i \in I_j$ and $i \notin I_j$. These matrices, though, are zero when the 3D point i cannot be seen by the texel swath j or when that projection point cannot be found. This reduces computation significantly.

In addition, the vector $\epsilon_{ij} = \mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij}$ is calculated. As mentioned before, this value can take on two forms. For $i \in I_j$,

$$\epsilon_{ij} = \begin{bmatrix} (x_{ij} - \hat{x}_{ij}) \\ (y_{ij} - \hat{y}_{ij}) \\ (\lambda_{ij} - \hat{\lambda}_{ij}) \end{bmatrix} \tag{3.11}$$

where x_{ij} and y_{ij} are the calibrated projection values and λ_{ij} is the measured range. In contrast, for $i \notin I_j$

$$\epsilon_{ij} = \begin{bmatrix} (x_{ij} - \hat{x}_{ij}) \\ (y_{ij} - \hat{y}_{ij}) \\ 0 \end{bmatrix} \tag{3.12}$$

where x_{ij} and y_{ij} are the projected points determined by Algorithm 2 and there is no measured range. If Algorithm 2 does not find a projection match for a particular i-j combination, the error vector is simply $\epsilon_{ij} = \mathbf{0}$. Relating ϵ_{ij} to (3.5) , the system error can be written as $\mathcal{E}^2 = \sum_i \sum_j \epsilon_{ij}^T \Sigma_{ij}^{-1} \epsilon_{ij}$ where

$$\Sigma_{ij}^{-1} = \begin{bmatrix} \frac{1}{\sigma_I^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_I^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_\lambda^2} \end{bmatrix} \tag{3.13}$$

for $i \in I_j$ and

$$\Sigma_{ij}^{-1} = \begin{bmatrix} \frac{1}{\sigma_{\bar{I}}^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_{\bar{I}}^2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

for $i \notin I_j$. These scatter matrices ensure the dimensional analysis of the system.

After these computations, intermediate matrices are computed for all i and all j

$$\begin{aligned} U_j &= \sum_i A_{ij}^T \Sigma_{ij}^{-1} A_{ij} \quad & V_i &= \sum_j B_{ij}^T \Sigma_{ij}^{-1} B_{ij} \\ W_{ij} &= A_{ij}^T \Sigma_{ij}^{-1} B_{ij} & Y_{ij} &= W_{ij} V_i^{*-1} \\ \epsilon_{\mathbf{a}_j} &= \sum_i A_{ij}^T \Sigma_{ij}^{-1} \epsilon_{ij} \quad & \epsilon_{\mathbf{b}_i} &= \sum_j B_{ij}^T \Sigma_{ij}^{-1} \epsilon_{ij}. \end{aligned} \tag{3.15}$$

The * symbol on a matrix means its diagonal elements are multiplied by $1 + \Lambda$ (where Λ is the step-size parameter, usually initialized at about $\Lambda = 0.001$). Note the difference between the summation $\sum_i$ and the scatter matrix Σ_{ij}^{-1}. For any points i not viewed from

image j, the associated matrices are simply left out of the sums, as points that cannot be seen do not contribute to the total system error.

After these intermediate matrices are computed, the parameter perturbation $\delta_{\mathbf{a}}$ is determined by solving

$$S\delta_{\mathbf{a}} = (\mathbf{e}_1^T, ..., \mathbf{e}_M^T)^T \tag{3.16}$$

where $\delta_{\mathbf{a}} = (\delta_{\mathbf{a}_1}^T, ..., \delta_{\mathbf{a}_M}^T)^T$, and S is an $M \times M$ block matrix whose elements consist of

$$S_{jj} = -\sum_i Y_{ij} W_{ij}^T + U_j^*, \tag{3.17}$$

$$S_{jk} = -\sum_i Y_{ij} W_{ij}^T \text{ for } j \neq k, \tag{3.18}$$

and

$$\mathbf{e}_j = \epsilon_{\mathbf{a}_j} - \sum_i Y_{ij} \epsilon_{\mathbf{b}_i} \tag{3.19}$$

for $j = 1, ..., M$.

Once $\delta_{\mathbf{a}}$ is computed, each $\delta_{\mathbf{b}_i}$ is calculated using

$$\delta_{\mathbf{b}_i} = V_i^{*-1}(\epsilon_{\mathbf{b}_i} - \sum_j W_{ij}^T \delta_{\mathbf{a}_j}). \tag{3.20}$$

Each perturbation vector $\delta_{\mathbf{a}_j}$ and $\delta_{\mathbf{b}_i}$ are added to $\mathbf{a}_j$ and $\mathbf{b}_i$ to form $\hat{\mathbf{a}}_j$ and $\hat{\mathbf{b}}_i$, respectively. These updated parameters and data points are then used to test the system error $\mathcal{E}^2$ and determine if this error has decreased with the new parameters and points.

If the squared-error has decreased, then Λ decreases by a factor of ten, and the next iteration takes place with the updated parameters $\mathbf{a}_j = \hat{\mathbf{a}}_j$ and points $\mathbf{b}_i = \hat{\mathbf{b}}_i$. If the squared-error has not decreased, then Λ increases by a factor of ten, and the parameters and points remain unchanged.

These iteration steps occur until the change squared-error in each successive step drops below some threshold, at which point a minimum is determined to have been found.

Computational and Convergence Improvements

Because of the potential for a large number of both data points and texel swaths to be adjusted in the Sparse LMA, there are several ways in which convergence and computation can be improved.

If a particular swath j does not see a particular 3D point i, then there is no need to calculate the partial derivative relationship A_{ij}, B_{ij}, and ϵ_{ij} as these will be zero by virtue of having no direct relation to one another. The zero matrices are flagged such that they are omitted from the sums in the algorithm using a simple test rather performing the matrix multiply-add.

Another method to decrease computational time is to window the points that a swath can see. If the number of swaths seen by a given swath is restricted to only in nearby swaths, then the number of computations is reduced – both in the LMA and in Algorithm 2.

The error in the LMA can reach a point when it does not decrease significantly with each successive iteration, following the concept of diminishing returns. The computation costs can outweigh any error mitigation. This occurs typically after the camera parameters have been optimized, but the many 3D points are moving very small distances to reduce the error by a small amount. In other words, there is not much gain in each additional iteration. The completion change threshold should be set above these steady-state changes.

One other consideration not inherent in the LMA is the fact there may be outliers present in the data. Because the LMA minimizes the sum of squared error, outliers can contribute significant amounts to the error. Rather than using squared error as a cost function, other cost functions can be implemented. One simple alternative is that of Blake and Zisserman [54]. After the error reaches a threshold, the error is saturated and set as a constant value. That is, this cost function limits the amount of error any outlier can contribute. Other cost functions are listed in Hartley and Zisserman [47, p. 616].

3.4.2 Incremental Optimization Approach

An alternative to performing bundle adjustment on the entire data set is to perform it incrementally. This is conducive to real-time computations, and, if real-time, would be a realization of Simultaneous Localization and Mapping (SLAM) concepts [55], as it determines location and the 3D landscape concurrently.

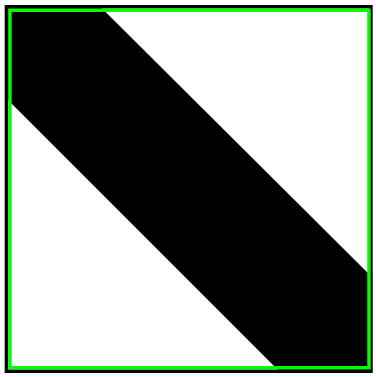
If the parameters for a small data set have been optimized, and an additional parameter is introduced into the data set, not many changes need to take place to optimize the additional parameter. Applying this to texel swaths, most of the changes will occur by placing the swath and its associated points in the correct locations. The points already optimized, especially those further away from the new data, will not move a significant amount. Thus, bundle adjustment can be performed by either doing bundle adjustment on the entire set of data each time a new swath is added, or by doing bundle adjustment on a subset of the data that is affected by the new swath.

This allows for some flexibility in the optimization. A graphical representation of the bundle adjustment ideas is shown in Fig. 3.5, which shows the concepts of using all the data for the optimization, adding the data incrementally, and optimizing using a sliding window.

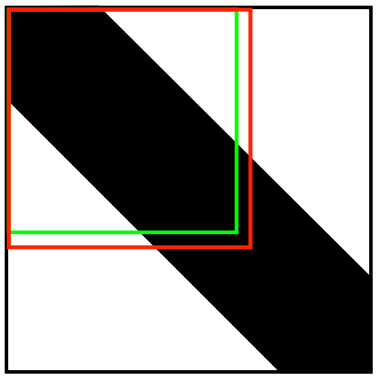
3.4.3 Seeding the Optimization

Many points in the system are nearly collinear, which makes triangulation difficult. Collinearity is shown in Fig. 3.6. There is greater area in which a point can be found when the cameras are close to being collinear. Sometimes errors obvious to the human eye amount to nothing more than noise to a computer. The LMA is sensitive to the initial model parameters, and because of collinearity, these parameters must be close to being optimized. The optimization is initialized with the coarse estimates of the camera locations and attitudes and the 3D points transformed from their captured coordinate system into the world coordinate system using the coarse estimates.

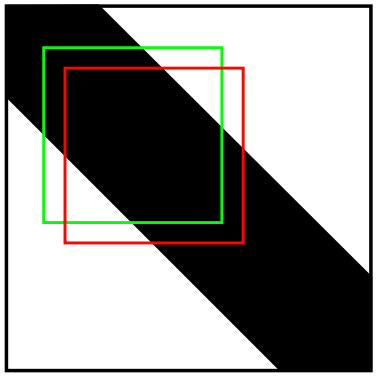
The fundamental matrix from texel swath to texel swath can be incorporated in the process to give potentially better estimates of the camera locations and attitudes for an optimization seed. The fundamental matrix for a given swath pair can be computed from



(a)



(b)



(c)

Fig. 3.5: Types of system bundle adjustment. The dark band in each square represents the diagonal structure of the Jacobian matrix. The green and red outlines show the data that is optimized in each successive increment. (a) The entire data set is optimized at once. (b) The entire data set is optimized with each successive set of measurements. (c) The optimization is performed on the subset of data which is most affected by the additional information while leaving the most previous data unchanged, i.e. sliding a window along the band.

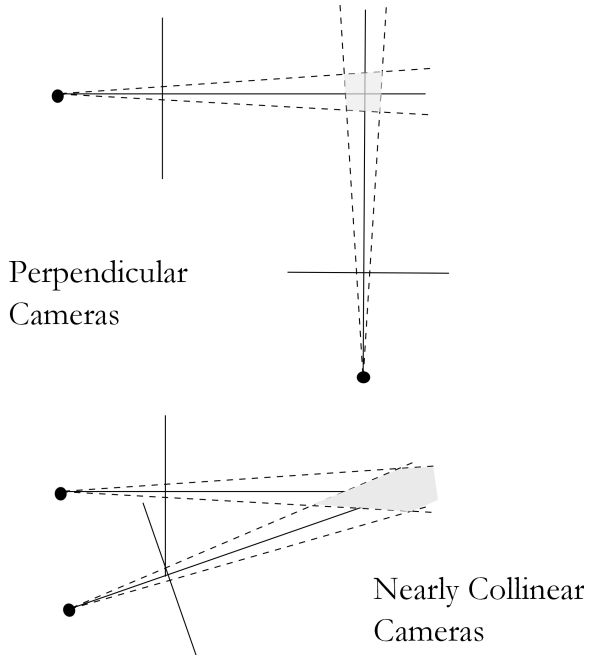


Fig. 3.6: Showing the collinearity problem. The shaded areas show the intersection of an error frustrum from each camera.

corresponding points in adjacent images, using the same putative correspondences when finding the homography in Algorithm 2. From this, relative rotation and translation between swaths can be extracted as explained in Section 2.8.2. These rotations and translations from swath to swath can be concatenated from one to the next. The location and attitude of each successive swath can be determined by concatenating the information from one to the next.

3.5 Textured Digital Elevation Model Creation

The desired end-product of the algorithm is to form a textured digital elevation model (TDEM). The Sparse LMA Bundle Adjustment gives a point cloud upon completion. Delaunay triangulation [56] (based on the x- and y-coordinates) can be used to form a surface over the point cloud. This surface can be textured or colored for the desired application. One way to do this is to do an orthorectified texture. An orthorectified image is one such that the camera is placed at ∞. Essentially, each pixel on the image is observed nadir, rather than at some skew angle inherent in the pinhole model.

If an orthorectified image is desired, all that must be done to project the 3D points onto the image plane is remove their z-coordinates (or whichever axis lies along the nadir direction). This leaves a 2D plane (similar to the normalized image plane in the pinhole model) with a collection of 2D points. At this point, a digital image can be overlaid on these points. The number of pixels per unit length can be set at the desired value. For this research, the number is the same as the camera calibration focal distance (see Section 2.4.3) to maintain simplicity. Once this is determined, the pixel values need to be assigned pixel-by-pixel.

This proves to be an involved task. Rather than worrying about the orientation of the Delaunay triangles to determine which texel swath can best see the triangle, it is assumed that for a given pixel, the best texture to draw from is the texel swath that captured the 3D point closest to the given pixel. To put it another way: for a given pixel, find the closest projection of a 3D point in either pixel coordinates or normalized image coordinates. Once this 3D point i is found, then determine the texel swath j such that $i \in I_j$. Once the appropriate swath is found, its image must be interpolated to give the value for the given pixel.

One way to find the appropriate sampling location is to find a homography between the final texture and the given texel swath using the projected points found for the bundle adjustment. This is shown in Fig. 3.7.

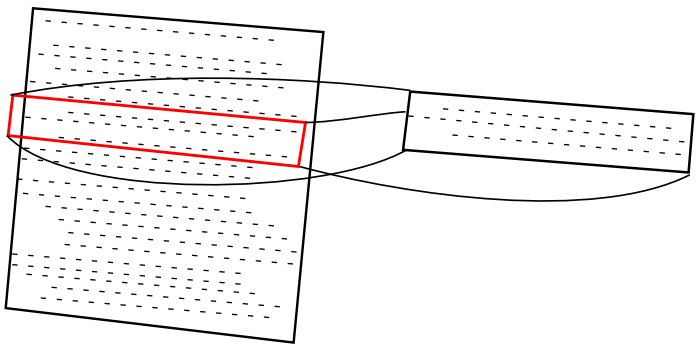


Fig. 3.7: Relationship between final texture (left) and a given texel swath (right). The dashes represent projected points which are used to determine a homography between the final texture and the given texel swath.

Once a homography is found, the sampling location is merely the coordinates of the pixel in the final texture transformed into the image swath and interpolating the appropriate pixels. There can be a problem using the homography assumption as the scenes may not be planar. The homographies between each texel swath and the final texture need only be computed once at the beginning of the algorithm. There is no blending in this process, which can cause some stitching seams.

Once the texturing is finished, the combined point cloud and texture can be considered a textured digital elevation model.

Chapter 4

Experimental Results

The swath-registration algorithm described in Chapter 3 was tested by gathering data using a short-range texel camera and a model landscape. The processing was done in C++ on a desktop computer. These registration results are analyzed with reference to a full-frame texel image of the scene. Section 4.1 describes the data set acquisition using the texel camera. Section 4.2 describes effectiveness of image processing concepts, and Section 4.3 describes the registration results for the process.

4.1 Texel Swath Acquisition

Because the second-generation handheld texel camera is a short range sensor, the gathering of texel swaths took place in the lab with a model landscape. The cardboard landscape included a large green field, an elevated plateau, a steep drop off, and some small hills, in addition to a blue-painted body of water. This landscape was ideal as it allowed for the texel camera to be placed about a meter away, and moved at tiny increments of a few millimeters at a time, simulating "flight" of a small UAV. The AHRS mounted on the texel camera recorded the attitude of each capture. A meterstick was used to measure the location of each capture. This setup is shown in Fig. 4.1a. In this chapter, the position and attitude information will be called the GPS/AHRS data regardless of how it is measured. An example EO image of the landscape captured from the texel camera is shown in Fig. 4.1b.

If this laboratory setup corresponds to a small UAV flight altitude of 300 meters, and texel swaths are captured approximately every 5 millimeters in the lab about one meter away from the landscape, this corresponds to swaths being captured every 1.5 meters in an actual flight. If the small UAV moves at 30 meters per second, the acquisition rate for the

swaths is about 20 Hertz. This is near-video rate. If a higher shot density on the ground is desired, the acquisition rate would be greater.

In order to better simulate the circumstances in which actual texel swaths are gathered (limited throughput for high image data rate), the EO image is trimmed so that only a region of imagery around the ladar points is kept, making one dimension much larger than the other dimension in the image. This is shown in Fig. 4.2. This allows for high-resolution data to be captured at video rate, but keeping only the most valuable parts of the imagery.

4.2 Image Processing Analysis

(a) (b)

Fig. 4.1: Data set acquisition. (a) Texel camera and landscape. (b) An example untrimmed EO image captured using the texel camera.

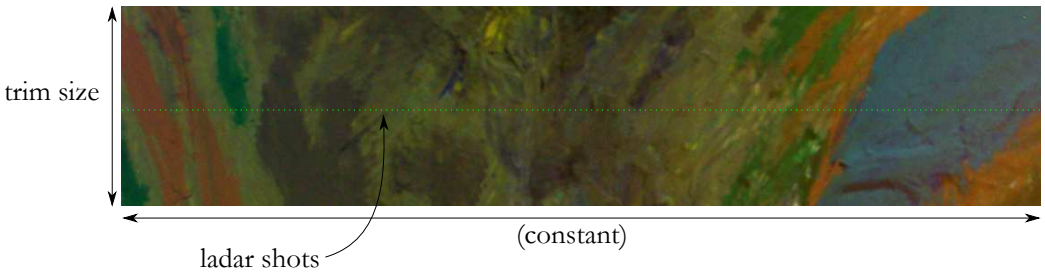
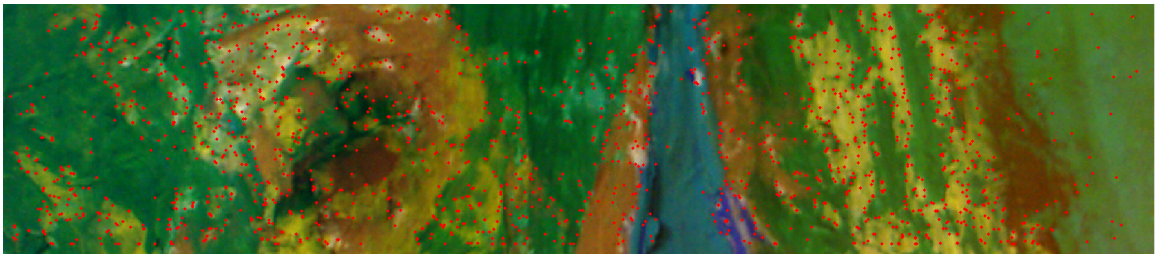


Fig. 4.2: The dimension of the image being trimmed about the ladar shots.

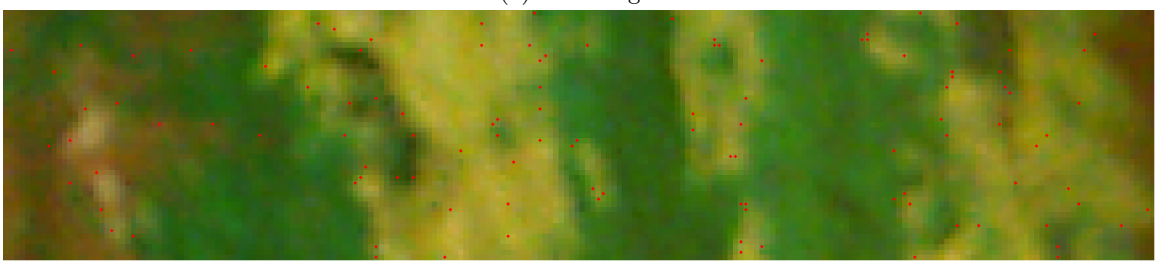
4.2.1 Harris Feature Point-Finding

Harris points are found in the images to determine matching points to be used in a homography calculation. The Harris Corner Detector operates on gray-scale images, but the EO images from the texel camera are color images. To process the color images, the algorithm operates on each color plane separately and independently, then concatenates the results. The color space luminance-intensity-chrominance (YIQ) seems to give slightly better points than the color space red-green-blue (RGB). It was observed that the parameters used in the Harris Corner Detector were scene-specific and returned significantly different numbers of feature points in images of different scenes.

An example image with Harris feature points (found in the YIQ color space) is shown in Fig. 4.3. These points perform well for the homography-RANSAC calculations after being matched using NCC.



(a) Full image.



(b) Enlarged to show detail.

Fig. 4.3: An image with Harris feature points highlighted in red. These feature points were determined with the response given by Noble. The points tend to cluster around edges, where there is a large change in color, or otherwise busy areas. Bland areas show a lack of Harris features.

4.2.2 Fundamental Matrix Calculation

The fundamental matrix calculations were based on both the measured rotation and translation between two texel swaths as well as correpsonding points between two EO images. This allows for the search area to be limited when finding putative correspondences as well as recovering camera position and location, respectively.

Fundamental Matrix from Measured Location and Attitude

This calculation is straightforward, but the fundamental matrix is only as accurate as the measurements made. An example pair of EO images, with image points and epipolar lines highlighted using the fundamental matrix calculated from GPS/AHRS measurements, is shown in Fig. 4.4. The epipolar lines are vertical because the primary movement between the two swaths is a translation in the "up" direction. In the figure, note how the epipolar lines do not intersect with the associated points. This is due to the error in the translation and/or the rotation measurements.

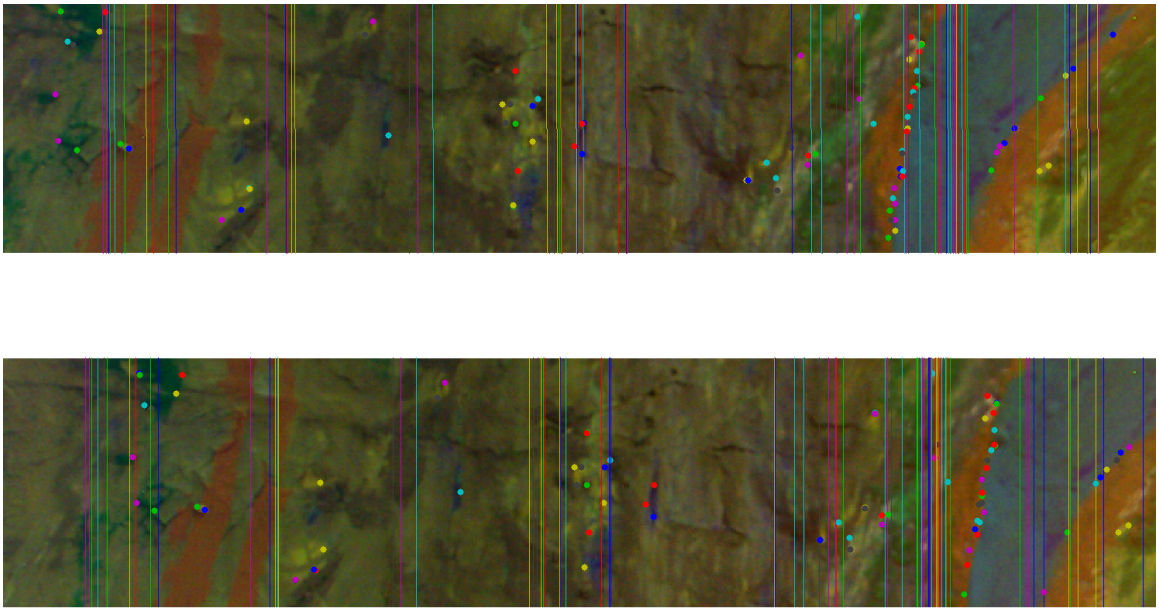


Fig. 4.4: Putative correspondences and associated epipolar lines using the fundamental matrix derived from the measured R and $\mathbf{t}$. Each color represents a given point-epipolar line pair.

To quantify this error, the distance to the scene and the intrinsic camera calibration matrix is used. If the error can be measured in pixel distance, this can be converted to a distance in the normalized image plane by dividing by the focal distance. In the texel camera case, an error of 40 pixels in normalized image plane distance is $\frac{40}{1361} = 0.0294$. This distance can be converted to Euclidean distance by multiplying by the range from the camera to the scene. The range in this example is about one meter, so this corresponds to a Euclidean distance of 2.94 centimeters. If this error is entirely due to translation, then the camera translation error is 2.94 centimeters. If the error is entirely due to rotation, the error is the angle corresponding to an arc length (for small angles) of 2.94 centimeters at the given range. In this case, the error is about 1.7 degrees (assuming a one meter range, $S = r\theta$, $r = 1$).

This method of determining the fundamental matrix helps determine which points are putative correspondences. If the distance to the epipolar line is greater than a threshold, there is no need to use correlation to test if it matches because it is assumed not to be a candidate.

Fundamental Matrix from Corresponding Points

Computing the fundamental matrix from corresponding points works well using the Eight-Point Algorithm with RANSAC in most cases. However, due to the nature of the texel swath imagery the Eight-Point Algorithm does not recover rotation and translation reliably.

An example set of texel swaths showing matching points as well as corresponding epipolar lines calculated from matching points is shown in Fig. 4.5. Notice how the epipolar lines seems to converge towards a point more quickly in Fig. 4.5 than in Fig. 4.4. This is because the epipole is closer to the principal point in Fig. 4.5, probably due to camera rotation. It was determined that RANSAC does not always choose the correct model, because there is little difference between a good model and a bad model due to the nature of the corresponding points and how they are scattered about in an image.

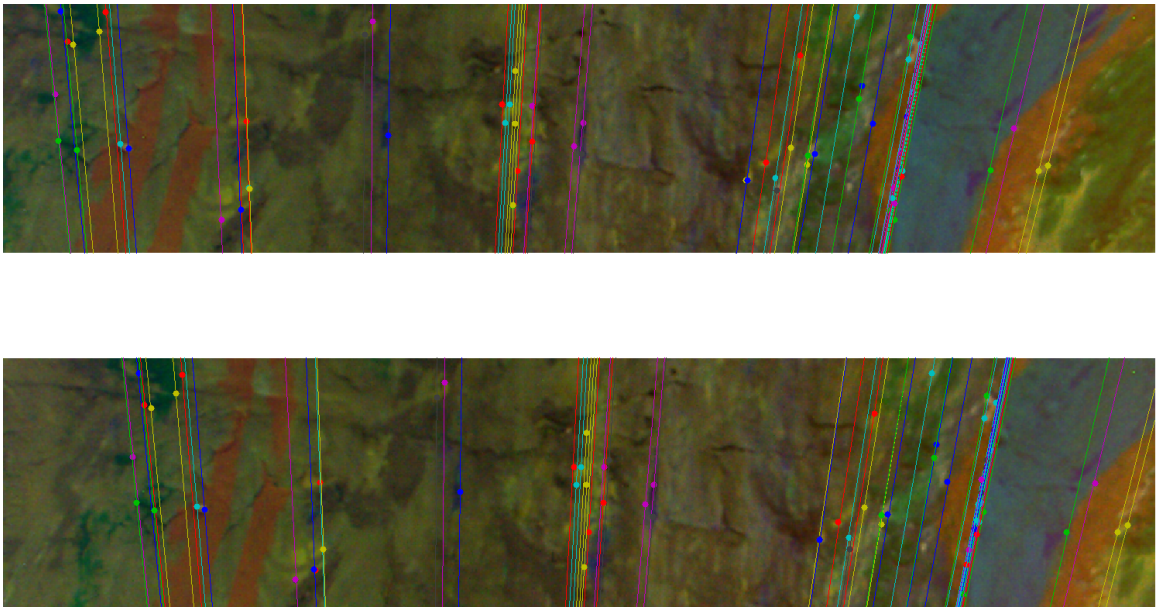


Fig. 4.5: Matching points and corresponding epipolar lines from the fundamental matrix found using corresponding points.

Imagine that the images in Fig. 4.5 were taken from an aerial vehicle with the texel camera looking straight down with the aerial vehicle moving perpendicular to the width of the swath. The yaw rotation is well-defined by the corresponding points. The roll rotation is also well-defined, with many points are spread out in the left-right direction of the image. However, the pitch rotation is not well-defined by the points, due to the image trim width being small. This means there are ambiguities between small pitch movements and small forward/backward translations. This ambiguity is shown in Fig. 4.6. Each pair of corresponding points has a great influence on this ambiguity, and some pairs of points chosen in RANSAC will favor rotation over translation or vice-versa.

Thus, recovering rotation and translation using the corresponding point method is not reliable when corresponding points are primarily along one direction in an image. This is the problem for texel swaths.

4.2.3 Finding Projection Points

Using the experimental setup, finding the projection points performed well given the resolution of the EO images. The window size for the correlation needed to be fairly large

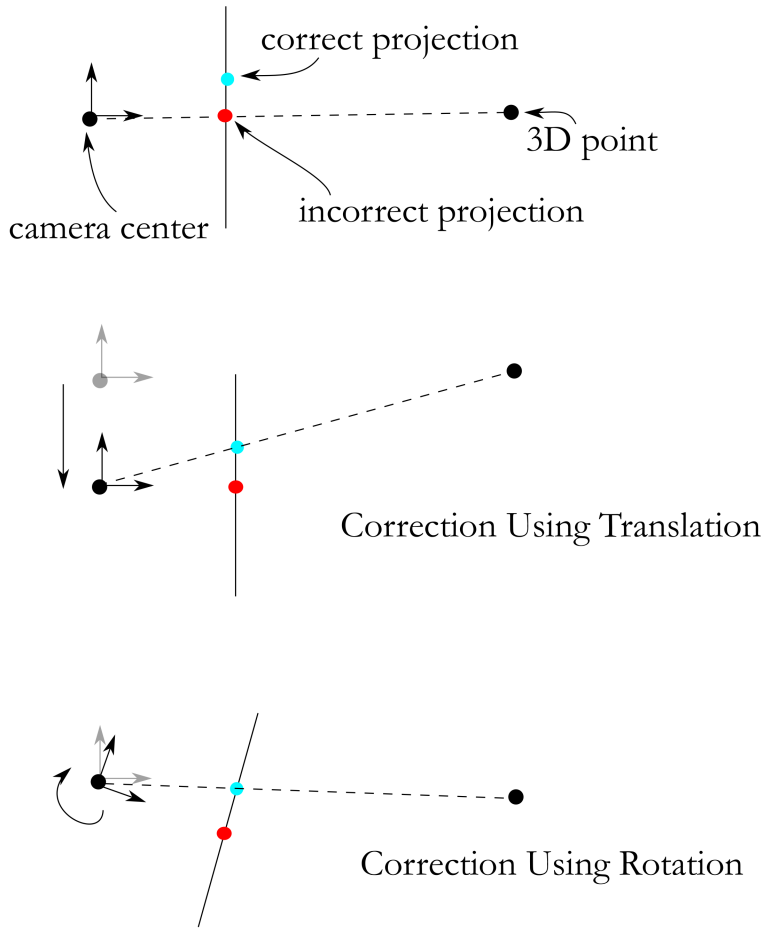


Fig. 4.6: Ambiguity of small rotations and translations. The top view of a camera (with its coordinate system), a 3D point, and projections on the image plane are shown in each illustration. The top illustration shows the original problem with a 3D point and its unmatched projection (or epipolar line). The middle illustration shows the projection error removed using translation of the camera and the bottom illustration shows the error removed using rotation of the camera.

(around 30×30 pixels) in order to catch enough unique imagery to give a good match. The search area for this correlation window was about 20×20 pixels due to the homography being able to find a good starting point for the search.

An example of projection points found using this algorithm is shown in Fig. 4.7. This figure illustrates that a given texel swath can only see points from a limited number of adjacent swaths. The number of swaths an image can see is related to how wide the image is "trimmed" in the up-down direction of the image, as well as how far apart the images were taken.

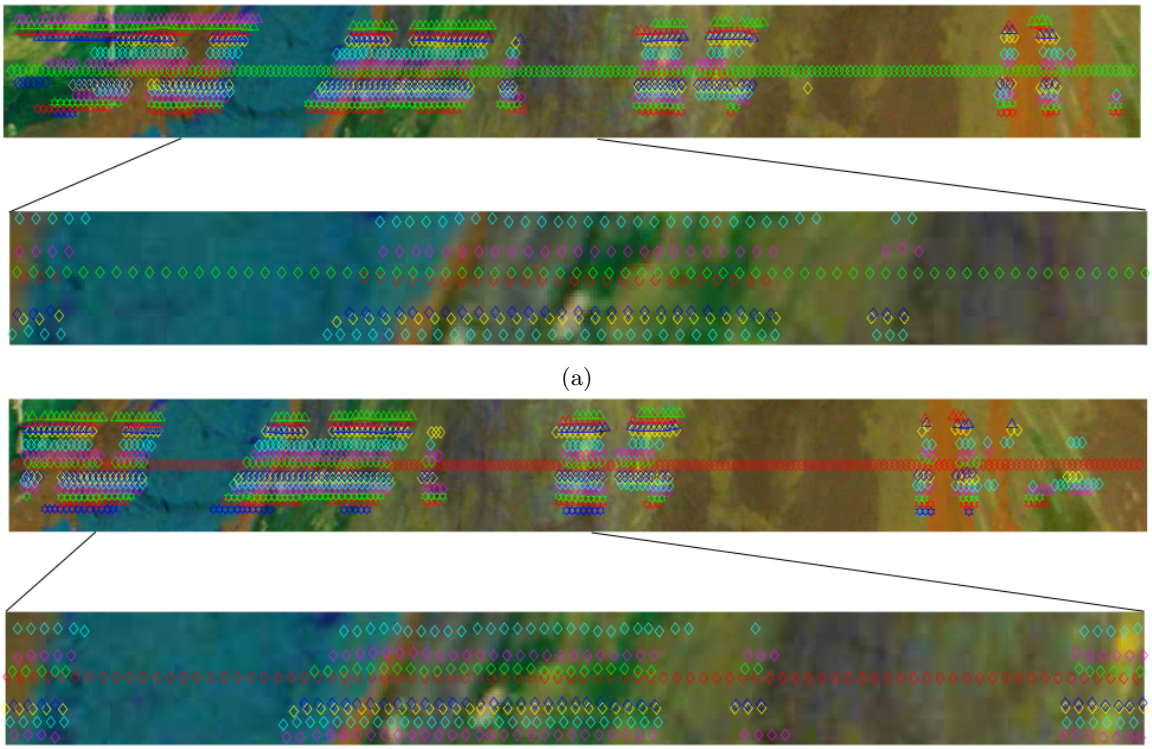
Projection points were not found in bland areas of the image, due to the correlation score not being high enough to give a unique identifier. Most of the projection points were found in image areas with color changes. In Fig. 4.7 the projection points can be found around areas with significant color changes in the imagery, like the edges of the blue, green, and brown patches.

For the data sets gathered for this research, NCC worked acceptably well to find projection points. For other data sets when most of the imagery is bland, other techniques may need to be used to match image patches.

4.3 Swath Registration Results

This section shows the results of the registration compared to a full-frame texel image of the same scene. The full-frame texel image is considered to be ground truth. An alternative method of comparison could use surveyed points. However, these data sets are gathered with a short-range texel camera using a model cardboard landscape with no surveyed points. Hence, a full-frame texel image is a reasonable alternative to surveyed points.

Three data sets are registered and analyzed in this section. The open source C++ matrix library, Armadillo [57], was used to implement the LMA. All computations were done in C++.



(a)

(b)

Fig. 4.7: Image points projected from ladar points found using correlation in adjacent swaths. Colors correspond to the points projected from ladar points in the current swath and swaths before and after the current swath. Each color corresponds to points from one swath. The line of points down the center of each swath are calibrated 2D points projected from the ladar points acquired with that swath. (a) First swath and an enlarged area. (b) Second swath and an enlarged area.

4.3.1 Metric for Analysis

Metrics for analyzing the accuracy of a TDEM are not well-defined and there are multiple ways for analysis. One method is to measure the distance between two points on the model and two corresponding surveyed points as a reference to get an idea of how distances on the model compare to distances in the real world. Another method is to evaluate areas of polygons created by surveyed points. The cardboard landscape, however, does not have any surveyed points. Thus, in order to be able to create a reference, a full-frame texel image is used for the surveyed points.

Several 3D points can be selected on both the full-frame model and the registered model. These points can be permuted to find the distances from each point to every other point. For this analysis, $n = 10$ points are chosen, which results in $\binom{10}{2} = 45$ distances, and values for the error mean, the error standard deviation, and the root-mean-squared error between corresponding distances on a full-frame texel image and the registered TDEM for each case are shown.

The p^{th} 3D point on the full-frame model in the analysis is given as χ_p, with its corresponding point in the registered model $\hat{\chi}_p$. The distance between χ_p and another point on the full-frame model χ_q is d_{pq}, and, similarly, the distance between the points $\hat{\chi}_p$ and $\hat{\chi}_q$ on the registered model is $\hat{d}_{pq}$. The error mean is given by $\mu = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j=i}^{n} (d_{ij} - \hat{d}_{ij})$. The other statistics are computed in a similar manner.

4.3.2 Data Set Analysis

Level Flight

This data set consists of moving the camera a few millimeters for each capture approximately one meter away from the model landscape, and contains 150 texel swaths. The camera maintained a relatively constant attitude, but a significant translation offset was introduced part-way through the data set.

The parameters used for this optimization include $\sigma_I = 0.001$ (1.36 pixels), $\sigma_{\bar{I}} = 0.002$ (2.72 pixels), and $\sigma_\lambda = 0.001$ m. No windowing was enforced, apart from that caused by the

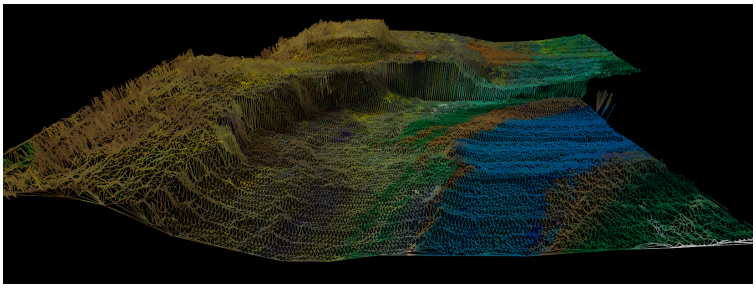
EO trimming. Convergence for the LMA was determined to be a 0.5% change in successive error, with a maximum number of iterations set at 100. The EO image width (trim size) was adjusted to determine the accuracy of the registration if fewer adjacent swaths are seen with limited imagery. The registration results are shown in Table 4.1. This data set analysis does not incorporate any fundamental matrix calculations.

The errors in the unregistered TDEM confirm the need for a registration algorithm to create an accurate and visually appealing TDEM. These errors make the model look noisy and jagged in a 3D view. In addition, the overlaid image has significant stitching seams.

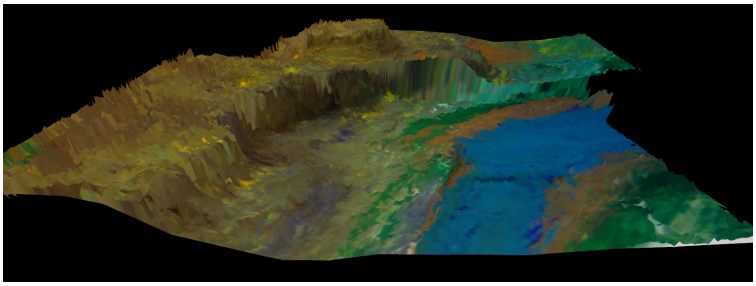
Figure 4.8 shows TDEMs both before the optimization and after the optimization using the 200 pixel trim size. The algorithm does very well for correcting the translation offset, especially when the EO image width is quite wide. Visually, the large translation offset is gone, and there are several minor adjustments to the other swaths making it visually appealing. As the quantities in Table 4.1 suggest, these resultant TDEMs are accurate using the full-frame model as a reference, and more accurate as the number of adjacent swaths seen increases. Figure 4.9 shows the full-frame texel image used as a reference.

Trim Size (pixels)	**Error Mean** (mm)	**Error Std Dev** (mm)	**RMS Error** (mm)	**Comments**
Before Registration	5.24	40.1	40	Includes a significant translation offset
200	0.690	4.08	4.10	Good registration, LMA converged relatively quickly
180	0.493	4.13	4.11	Good registration
160	1.02	4.40	4.47	Noticeable errors, but good registration overall
140	0.214	4.47	4.42	Noticeable errors, but good registration overall
120	8.10	16.0	17.8	Significant errors

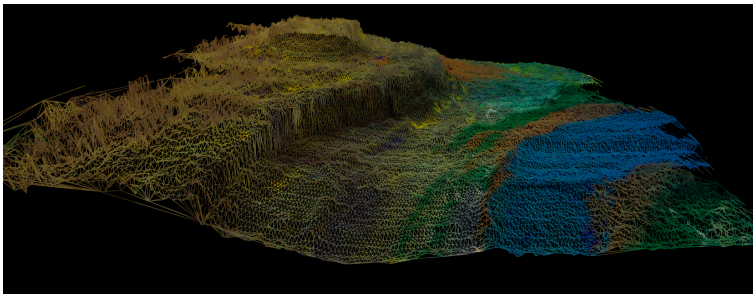
Table 4.1: Error for Various EO Widths for Level Flight Data Set



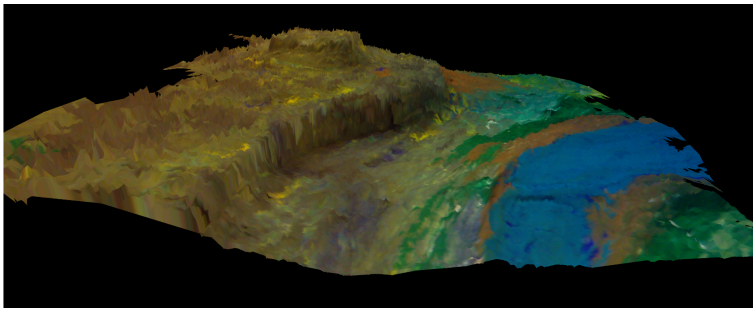
(a)



(b)

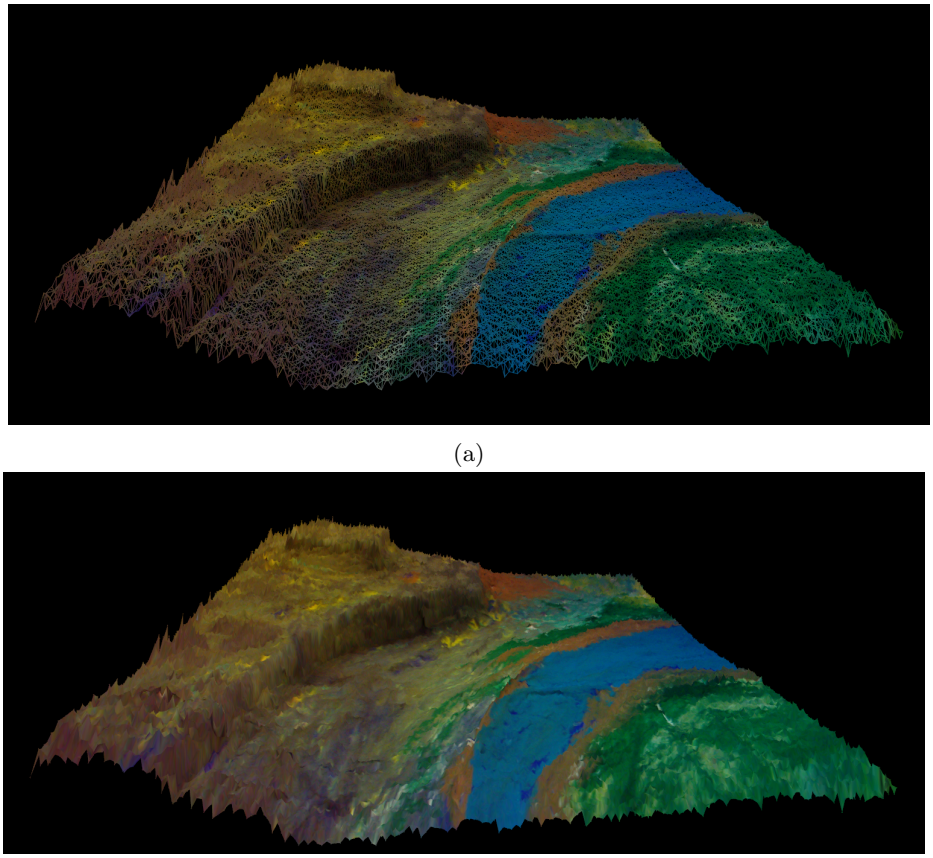


(c)



(d)

Fig. 4.8: Smooth flight data set registration. (a) Wire-frame before optimization. (b) TDEM before optimization. (c) Wire-frame after optimization. (d) TDEM after optimization.



(a)

(b)

Fig. 4.9: Smooth flight full-frame reference texel image. (a) Wire-frame. (b) Textured.

Turbulent Flight

This data set consists of moving the camera a few millimeters for each capture approximately one meter away from the model landscape and contains 177 texel swaths. The turbulence consists primarily of "roll" relative to the landscape. Figure 4.10 shows the approximate amount of roll present in the data set. Considering the camera field of view is approximately 40°, the nearly 30° peak-to-peak roll is extreme, demonstrating the robustness of the algorithm. This data set analysis does not incorporate any fundamental matrix calculations.

The parameters used for this optimization are the same as for Section 4.3.2. The results of these experiments are shown in Table 4.2. The errors in this experiment are greater than

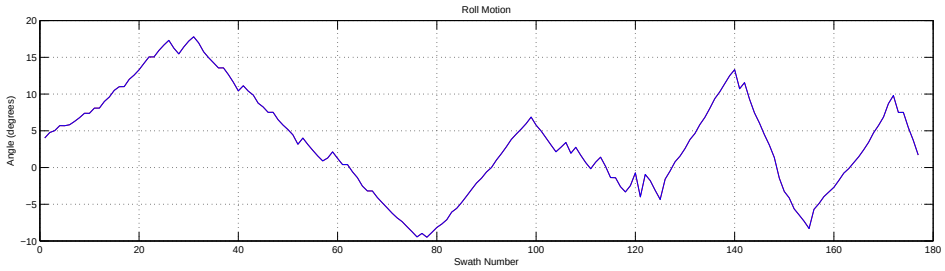


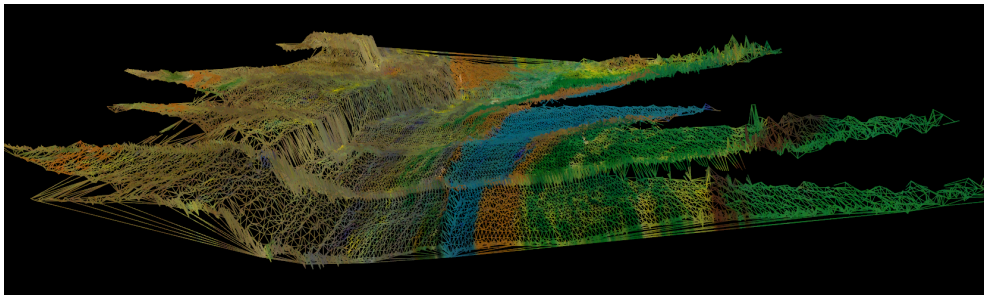
Fig. 4.10: Roll profile for the turbulent flight data set.

Trim Size (pixels)	**Error Mean** (mm)	**Error Std Dev** (mm)	**RMS Error** (mm)	**Comments**
Before Registration	-8.32	28.4	28.1	Contains significant roll
260	7.48	8.83	11.5	Good registration
240	7.86	8.67	11.6	Good registration
220	4.94	12.0	12.9	Good registration locally, but noticeable errors
200	4.71	12.6	13.4	Good registration locally, but noticeable errors
180	1.31	15.1	15.0	Good registration locally, but noticeable errors
160	1.05	13.1	13.0	Good registration locally, but noticeable errors

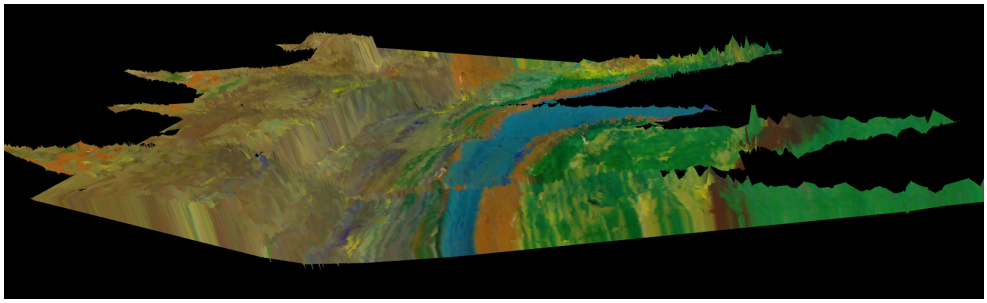
Table 4.2: Error for Various EO Widths for Turbulent Flight Data Set

in the level flight experiment, however, the registration improved the error compared to the unregistered TDEM.

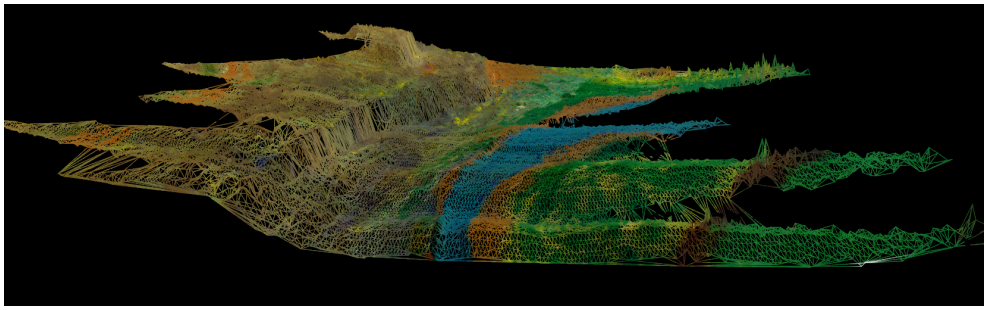
Figure 4.11 shows the TDEMs both before and after the optimization, using the 240 pixels wide trim size. The quantities in Table 4.2 show that the registration significantly improves the distance error variance from the case using only the GPS/AHRS data, despite the presence of the camera roll. There are, however, artifacts from the roll, especially when the trim size is small. These artifacts are the vertical movement of the swaths, making the flat parts of the landscape look bulged in some areas. However, the overlaid texture looks good.



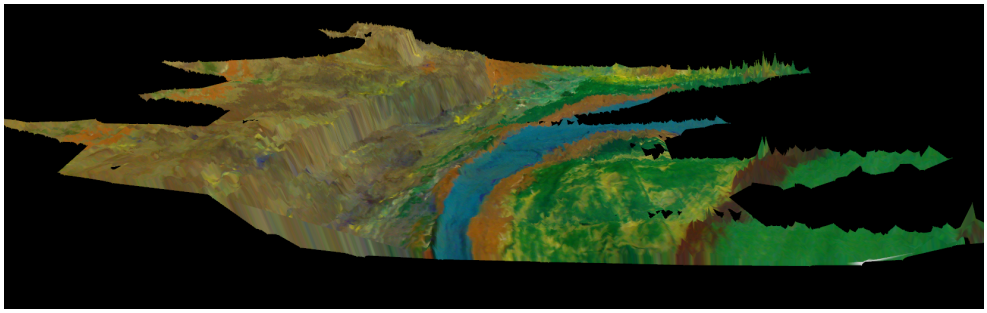
(a)



(b)



(c)



(d)

Fig. 4.11: Turbulent flight data set registration. Notice the significant roll motion. (a) Wire-frame before optimization. (b) TDEM before optimization. (c) Wire-frame after optimization. (d) TDEM after optimization.

Flight with a Turn

This data set consists of 134 texel swaths. A large number of the swaths represent a "turn" in the flight by about ninety degrees, as shown in Fig. 4.12. This data set is interesting in several aspects. First, the turn presents a slightly larger challenge when creating a homography between adjacent swaths than straight flight. This is because Harris feature points and NCC are not-rotationally invariant. However, the rotation from swath to swath in the turn amounts to only a few degrees, and the features and correlation do not seem to be affected significantly. This data set is also different because the capture-to-capture distances are larger than the other data sets; for a given swath trim size, fewer swaths can be seen. This necessitates a larger trim size for a better solution.

Also significant in this data set is the "variety" of scenes from image to image. The other data sets followed the model river, hovering over the edge of the model plateau. This data set starts out like this, but turns and moves over the field which consists different coloring and terrain. This affects the effectiveness of both feature-finding and NCC parameters tuned for the given scene.

The parameters used for this optimization include $\sigma_I = 0.001$ (1.36 pixels), $\sigma_{\bar{I}} = 0.002$ (2.72 pixels), and $\sigma_\lambda = 0.001$ m. No windowing was enforced, apart from that caused by the EO trimming. Convergence for the LMA was determined to be a 1% change in successive error, with a maximum number of iterations set at 100. The EO image trim size was adjusted to determine the accuracy of the registration if fewer adjacent swaths are seen. This experiment does not incorporate any rotation or translation extracted from the fundamental matrix, but uses the fundamental matrix calculated from known rotation and translation to help limit the number of putative correspondences that are compared when matching Harris feature points. A small area is searched about each feature point to find the best pixel match, which was not incorporated in the previous data sets. Table 4.3 shows the registration results of this data set.

These results show the distance error is reduced after the registration. The error mean remained relative constant while the error variations increased with decreasing trim size.

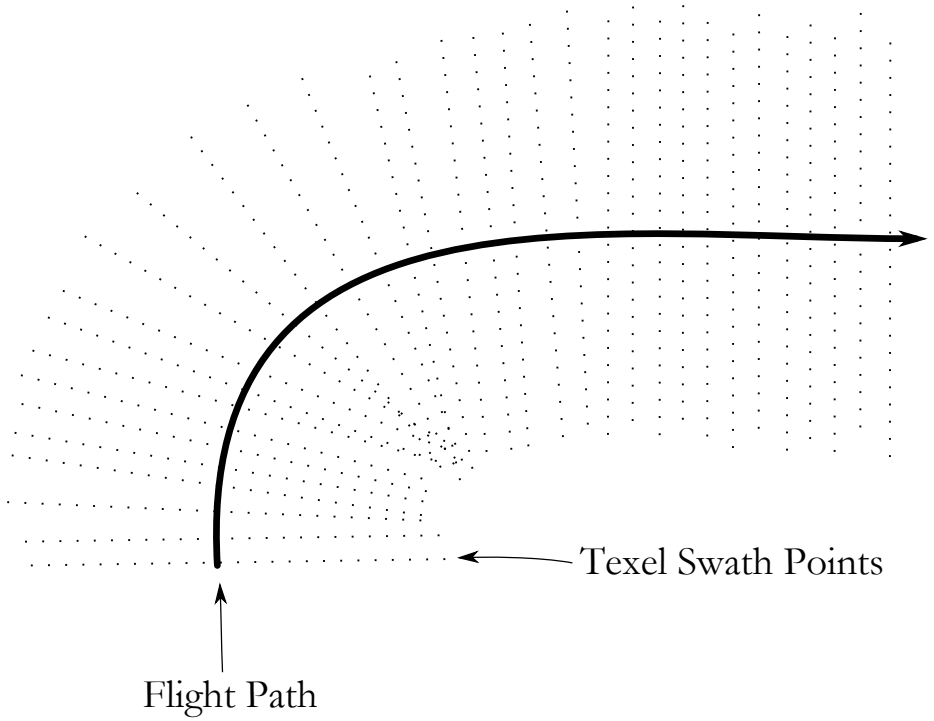


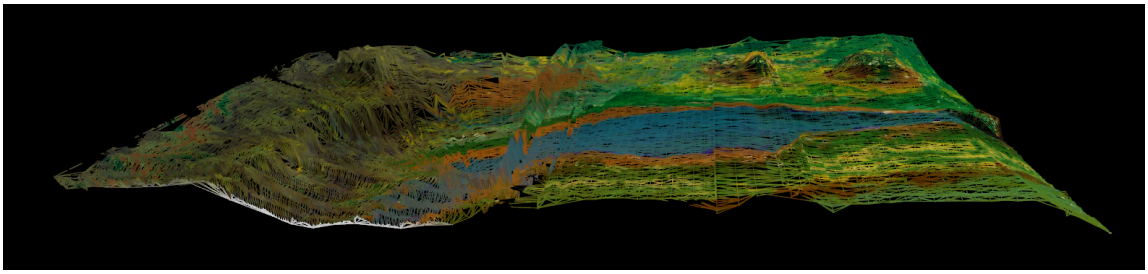
Fig. 4.12: Flight pattern for the turn data set. There is significant overlap on the inside of the turn.

Trim Size (pixels)	**Error Mean** (mm)	**Error Std Dev** (mm)	**RMS Error** (mm)	**Comments**
Before Registration	17.7	40.1	43.4	Contains vertical offset errors
241	-7.6	13.4	15.3	Good registration
221	-8.2	14.0	16.0	Good registration
201	-7.1	15.0	16.5	Good overall, but one area has deformation
181	-7.8	17.2	18.7	Good overall, but one area has deformation

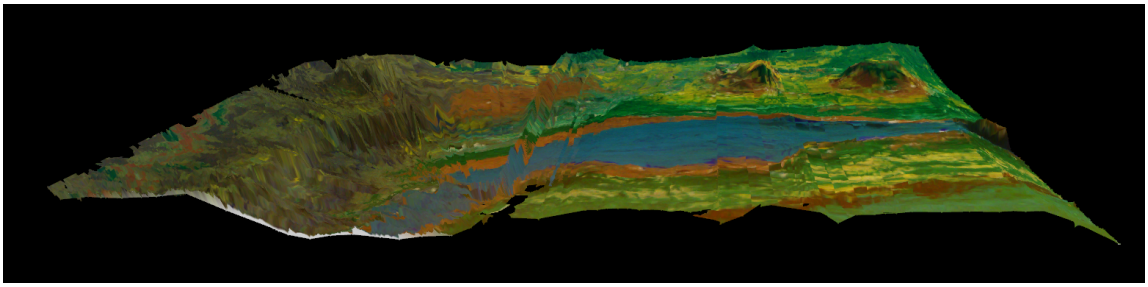
Table 4.3: Error for Various EO Widths for Turn Flight Data Set

The area with deformation occurs where the swaths are spaced farther apart, making image processing techniques less reliable for smaller sizes. Visually, as shown in Fig. 4.13, the registration improves over the unregistered data set significantly. There are some stitching

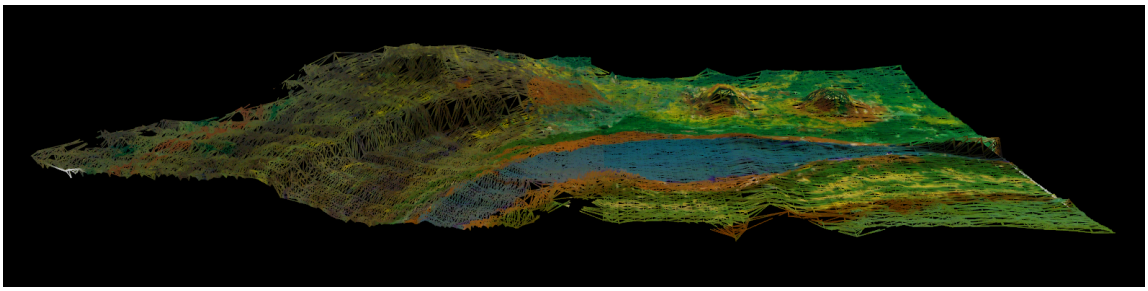
seams due to color differences in the images. There is a “noisy-looking” area on the inside of the turn, which is due to many 3D points present in the overlap.



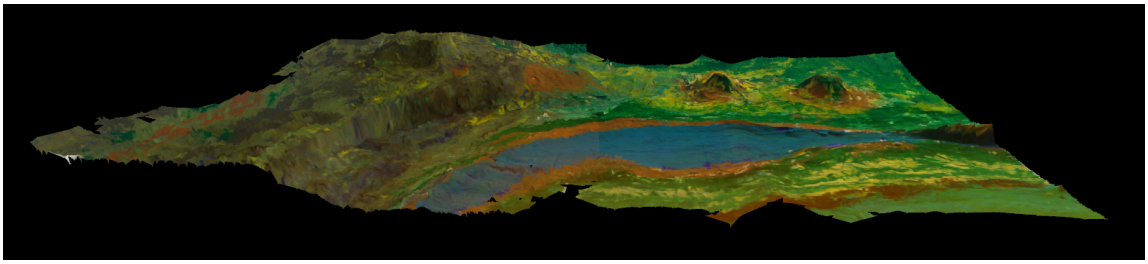
(a)



(b)



(c)



(d)

Fig. 4.13: Flight with turn data set registration. (a) Wire-frame before optimization. (b) TDEM before optimization. (c) Wire-frame after optimization. (d) TDEM after optimization.

4.3.3 Comparison to Photogrammetry

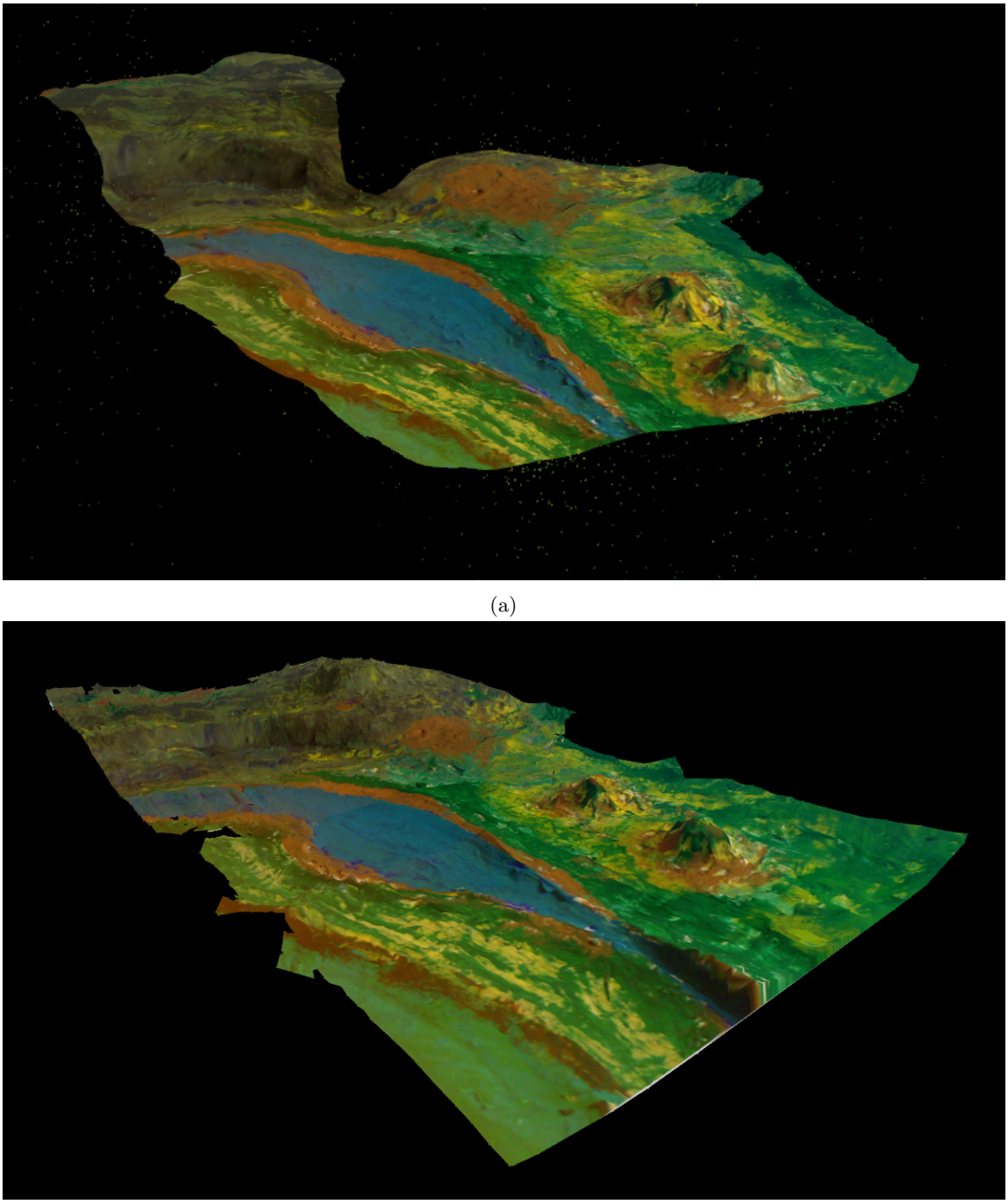
The results of the registration using photogrammetry alone was accomplished using freely downloadable programs Pix4Dmapper Discovery [58] and Visual SFM [59] and done on the turn data set. Visual SFM failed to produce an entire model using the trimmed images in each case. Pix4Dmapper Discovery was able to produce an entire model, but left some artifacts, as shown in Fig. 4.14. This photogrammetry-only technique did fairly well given the size of the images. The most notable difference is the large section missing on the plateau in Fig. 4.14a, as well as the edges of the data set. The surface shown is interpolated from points, but there are "free" 3D points that don't lie along the surface. However, there is curving of the surface, as compared in Fig. 4.15, but this could probably be offset to some degree using GCPs. There are no direct range measurements of the surface as there are with a ladar system, and any depth measurements are inferred from the image disparity.

The model created by Pix4Dmapper Discovery cannot be numerically evaluated in the same way as the quantities in Table 4.3 because correct scale is not present, and cannot be recovered without GCPs.

4.4 Discussion

This algorithm works well for the given data sets. It is no surprise that the registration works better with increasing swath width (more imagery information). The quality of the seed to the LMA seems to affect the optimization considerably, especially when the swaths are trimmed to a very small width. When there are small trim widths, there are more significant errors in the registration. This is not surprising. When a given swath can see only a small number of swaths, it is analogous to balancing a narrow board on a fulcrum; the solution becomes ill-conditioned in the sense that large rotation and translation movements cause only small changes in projection and range errors.

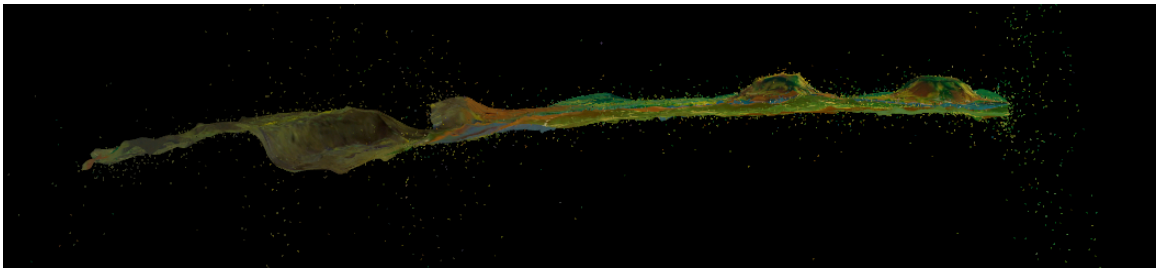
Each case took several hours to complete from start to finish. However, the code and process has not been thoroughly optimized, especially when creating the final texture and when finding projection points.



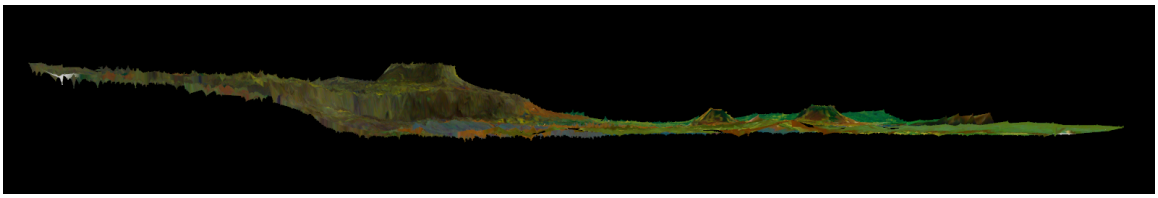
(a)

(b)

Fig. 4.14: Comparison of photogrammetry to texel swath optimization. (a) Pix4dmapper Discovery results. (b) Registered turn flight data set.



(a)



(b)

Fig. 4.15: Side comparison of photogrammetry to texel swath optimization.
(a) Pix4dmapper Discovery results from the side. (b) Registered turn flight data set from the side.

The fundamental matrix incorporation to estimate the image-to-image rotation and translation, as described in Section 2.8.2, and using these estimates to create a seed from cascading the estimates does not work well. This method seems to favor rotations over translations, probably due to the fact that only adjacent relatively small trim-size images are used. If this concept was extended to recovering camera pose using bundle adjustment photogrammetry techniques, it may provide a good seed. This would require much more computation, since the method would find a photogrammetry solution (such as that given by Pix4dmapper), followed by the algorithm presented in Chapter 3.

There are some stitching seams that can be seen in the TDEM after registration, but this is no surprise as the texturing was done in a rudimentary method. The texture looks much better in the registered TDEM than its unregistered counterpart, just due to the optimized point cloud.

A better metric, perhaps using surveyed points rather than a full-frame texel image, would be beneficial to understanding how well the algorithm reduces error. The registration is very pleasing for acceptably wide trim sizes.

Chapter 5

Conclusion and Future Work

The texel camera concept is valuable as it allows for the calibrated and simultaneous capture of range measurements and imagery. The preliminary results presented in this document are promising and open many avenues of research interest.

5.1 Conclusion

The quality of landscape reconstruction using the methods outlined in Chapter 3 is very pleasing. This method has the disadvantage of being computationally intense with both the image processing and LMA operations.

The registration for the "straight and level" data set registered quite well. This shows that when there is significant overlap and little or no perspective distortion from image-to-image, the algorithm performs without issue. With the "turbulent" data set, the image-to-image matching did not perform as well which seemed to cause some minor artifacts, such as bulging, in the final reconstruction. The "turn" data set proves that the algorithm works for a turn in the flight pattern as well as a variety of terrain. Overall, these results suggest that with wider swath size and with mild-to-moderate flight turbulence, the algorithm can find a useful textured digital elevation model.

One major drawback inherent in the problem is the near "collinearity" of the 3D points relative to the cameras in space. This drawback is magnified when each swath can see only a limited number of other swaths' points. This means that although locally it may register well, it does not register well globally. This can cause problems in the optimization, especially if the seed is bad.

The cost function is advantageous because it reflects how the points are captured in the texel camera: in projection-range coordinates. It is natural for 3D points to be represented

in this way when measured from a point. One disadvantage to the partial derivatives in the cost function is the computational complexity of the quaternion rotation and 3D-to-2D projection.

Traditional photogrammetry techniques would have trouble doing 3D reconstruction using the swath imagery alone, due to the dimensions of the images. However, this algorithm proves that with additional 3D information (camera pose and measured range) these problems can be mitigated.

5.2 Future Work

Typically areas of further work include optimizing the code to speed up the process. Often, a speed optimization does not improve the quality of results, so the focus of future work should be to improve the theory in the algorithm.

Further evaluation, involving other types of scenes, such as a canyon or cityscape should be investigated. After the algorithm is verified for these types of scenes, it should be tested on texel swaths taken from an actual aerial platform.

The optimization finds a local minimum that is far from the global minimum when the seed of the algorithm is not good. It is important that the information going into the optimization algorithm is good. Photogrammetry-only techniques could be used to do this in a more sophisticated way than using image-to-image fundamental matrices which failed in this research. Information about the actual location of the cameras can be used to find the correct scale for the camera location and attitude found using photogrammetry, which can then be used as a seed for the algorithm. Additional cost functions in the optimization should be investigated to help with outliers.

In addition, the fundamental matrix could be incorporated, not only in the pre-processing for the optimization but also in the optimization itself. All pairs of matching projections must meet the epipolar constraint, and the epipolar constraint is not limited to particular scene structures like a homography. In the cost function, the projection error is minimized. If the fundamental matrix was used, distance to the corresponding epipolar

line (Sampson distance) from the projection point could be minimized over the system (in addition to the range error).

Another important fact to consider in the optimization is that a particular camera (or image point, such as a GCP) may have coordinates known better than other cameras or points. Naturally, more emphasis should be placed on these measurements.

Also, the error model does not address the ambiguity of small rotations and small translations. That is, if a 3D point is moved a small amount, a small rotation and/or small translation of the camera can be found to minimize the projection error. Because there are many points in the system, it is hoped that the "average" correction will find the right balance between rotation and translation.

One factor that is often under-emphasized is the optimal number of points to compute homography and fundamental matrices, the threshold for NCC, the NCC window size, the Harris feature parameters, and so on. Because the algorithm deals with a large number of unique images in a given run, it is difficult to choose parameters that are good for the entire data set. A procedure should be developed that allows for adaptation of parameters during run-time to calculate the best parameter values for a given image or image pair.

Texturing the 3D point cloud is an involved task, and the method presented in Section 3.5 is rudimentary. Different stitching methods should be investigated, which may involve blending techniques and the relative pose of cameras to individual Delaunay triangles created from the point cloud.

Finally, this method relies heavily upon image-to-image processing concepts. If video processing concepts can be applied frame-to-frame, the selection of projection points may be more robust.

In summary, the strength of this algorithm lies in the optimization process, but the quality of its output depends directly on the success of finding corresponding projection points using image processing techniques. Overall, this algorithm is promising groundwork for future research by making automatic texel swath registration fast, accurate, and reliable.

References

[1] R. Hirokawa, D. Kubo, S. Suzuki, J.-i. Meguro, and T. Suzuki, "A small UAV for immediate hazard map generation," in *AIAA2007-2725. AIAA Infotech@ Aerospace 2007 conference and exhibit, Rohnert Park*, 2007, pp. 7–10.

[2] K. Anderson and K. J. Gaston, "Lightweight unmanned aerial vehicles will revolutionize spatial ecology," *Frontiers in Ecology and the Environment*, vol. 11, no. 3, pp. 138–146, 2013.

[3] K. Lim, P. Treitz, M. Wulder, B. St-Onge, and M. Flood, "Lidar remote sensing of forest structure," *Progress in physical geography*, vol. 27, no. 1, pp. 88–106, 2003.

[4] N. Yastikli, "Documentation of cultural heritage using digital photogrammetry and laser scanning," *Journal of Cultural Heritage*, vol. 8, no. 4, pp. 423 – 427, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1296207407001082

[5] E. P. Baltsavias, "A comparison between photogrammetry and laser scanning," *ISPRS Journal of photogrammetry and Remote Sensing*, vol. 54, no. 2, pp. 83–94, 1999.

[6] G. Conte, A. Kleiner, P. Rudol, K. Korwel, M. Wzorek, and P. Doherty, "Performance evaluation of a light-weight multi-echo lidar for unmanned rotorcraft applications," *UAV-g2013. The international archives of the photogrammetry, remote sensing and spatial information sciences, Rostock, XL-1 W*, vol. 2, pp. 87–92, 2013.

[7] G. Schut, "An analysis of methods and results in analytical aerial triangulation," *Photogrammetria*, vol. 14, pp. 16–33, 1958.

[8] F. Leberl and J. Thurgood, "The promise of softcopy photogrammetry revisited," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, no. Part B3, pp. 759–763, 2004.

[9] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[10] W. Förstner, "A feature based correspondence algorithm for image matching," *International Archives of Photogrammetry and Remote Sensing*, vol. 26, no. 3, pp. 150–166, 1986.

[11] T. Schenk, "Digital aerial triangulation," *International Archives of Photogrammetry and Remote Sensing*, vol. 31, pp. 735–745, 1996.

[12] T. Liang and C. Heipke, "Automatic relative orientation of aerial images," *Photogrammetric engineering and remote sensing*, vol. 62, no. 1, pp. 47–55, 1996.

[13] F. Ackermann and P. Krzystek, "Complete automation of digital aerial triangulation," *The Photogrammetric Record*, vol. 15, no. 89, pp. 645–656, 1997.

[14] D. M. Mount, N. S, and J. L. Moigne, "Efficient algorithms for robust feature matching," *Pattern Recognition*, vol. 32, no. 1, pp. 17 – 38, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320398000867

[15] C. Strecha, O. Küng, and P. Fua, "Automatic mapping from ultra-light UAV imagery," in *EuroCOW 2012*, no. EPFL-CONF-175351, 2012.

[16] M. Herman and T. Kanade, "The 3D mosaic scene understanding system: Incremental reconstruction of 3D scenes from complex images," 1984.

[17] R. T. Collins, C. O. Jaynes, Y.-Q. Cheng, X. Wang, F. Stolle, E. M. Riseman, and A. R. Hanson, "The ascender system: automated site modeling from multiple aerial images," *Computer Vision and Image Understanding*, vol. 72, no. 2, pp. 143–162, 1998.

[18] O. Küng, C. Strecha, P. Fua, D. Gurdan, M. Achtelik, K.-M. Doth, and J. Stumpf, "Simplified building models extraction from ultra-light UAV imagery," *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3822, p. 217, 2011.

[19] C. Strecha, R. Zoller, S. Rutishauser, B. Brot, K. Schneider-Zapp, V. Chovancova, M. Krull, and L. Glassey, "Terrestrial 3D mapping using fisheye and perspective sensors."

[20] T. Rosnell and E. Honkavaara, "Point cloud generation from aerial image data acquired by a quadrocopter type micro unmanned aerial vehicle and a digital still camera," *Sensors*, vol. 12, no. 1, pp. 453–480, 2012.

[21] A. S. Laliberte, C. Winters, and A. Rango, "A procedure for orthorectification of sub-decimeter resolution imagery obtained with an unmanned aerial vehicle (UAV)," in *Proc. ASPRS Annual Conf*, 2008, pp. 08–047.

[22] J. Rodriguez and J. Aggarwal, "Matching aerial images to 3-D terrain maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 12, pp. 1138–1149, Dec 1990.

[23] A. Wehr and U. Lohr, "Airborne laser scanning — an introduction and overview," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2, pp. 68–82, 1999.

[24] G. Tao and Y. Yasuoka, "Combining high resolution satellite imagery and airborne laser scanning data for generating bareland DEM in urban areas," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 30, 2002.

[25] M. Lim, D. N. Petley, N. J. Rosser, R. J. Allison, A. J. Long, and D. Pybus, "Combined digital photogrammetry and time-of-flight laser scanning for monitoring cliff evolution," *The Photogrammetric Record*, vol. 20, no. 110, pp. 109–129, 2005.

[26] D. Leckie, F. Gougeon, D. Hill, R. Quinn, L. Armstrong, and R. Shreenan, "Combined high-density lidar and multispectral imagery for individual tree crown analysis," *Canadian Journal of Remote Sensing*, vol. 29, no. 5, pp. 633–649, 2003.

[27] E. K. Forkuo and B. King, "Automatic fusion of photogrammetric imagery and laser scanner point clouds," *International Archives of Photogrammetry and Remote Sensing*, vol. 35, pp. 921–926, 2004.

[28] L. C. Chen, T.-A. Teo, Y.-C. Shao, Y.-C. Lai, and J.-Y. Rau, "Fusion of lidar data and optical imagery for building modeling," *International Archives of Photogrammetry and Remote Sensing*, vol. 35, no. B4, pp. 732–737, 2004.

[29] P. Rönnholm, E. Honkavaara, P. Litkey, H. Hyyppä, and J. Hyyppä, "Integration of laser scanning and photogrammetry," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 39, pp. 355–362, 2007.

[30] H. Badino, D. Huber, and T. Kanade, "Integrating lidar into stereo for fast and improved disparity computation," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on.* IEEE, 2011, pp. 405–412.

[31] A. S. Gneeniss, "Integration of lidar and photogrammetric data for enhanced aerial triangulation and camera calibration," 2014.

[32] F. C. Nex and F. Rinaudo, "Lidar or photogrammetry? integration is the answer," *RIVISTA ITALIANA DI TELERILEVAMENTO*, vol. 43, no. 2, pp. 107–121, 2011.

[33] E. Mitishita, P. Debiasi, F. Hainosz, and J. Centeno, "Calibration of low cost digital camera using data from simultaneous lidar and photogrammetric surveys," *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, pp. 133–138, 2012.

[34] B. Boldt, S. Budge, R. Pack, and P. Israelsen, "A handheld texel camera for acquiring near-instantaneous 3D images," in *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*, Nov 2007, pp. 953–957.

[35] "Cold mirror—edmund optics," http://www.edmundoptics.com/images/catalog/3149_LA2.gif, accessed: 2015-01-22.

[36] S. E. Budge and N. S. Badamikar, "Calibration method for texel images created from fused flash lidar and digital camera images," *Optical Engineering*, vol. 52, no. 10,

pp. 103 101–103 101, 2013. [Online]. Available: http://dx.doi.org/10.1117/1.OE.52.10.103101

[37] J.-Y. Bouguet, "Camera calibration toolbox for MATLAB," 2004.

[38] A. Buchmann, "A brief history of quaternions and of the theory of holomorphic functions of quaternionic variables," *arXiv preprint arXiv:1111.6088*, 2011.

[39] T. Sakamoto, C. Nakanishi, and T. Hase, "Software pixel interpolation for digital still cameras suitable for a 32-bit MCU," *IEEE Trans. Consum. Electron.*, vol. 44, no. 4, pp. 1342–1352, 1998.

[40] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, 1997.

[41] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[42] C. Harris and M. Stephens, "A combined corner and edge detector." in *Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[43] J. A. Noble, "Descriptions of image surfaces," Ph.D. dissertation, University of Oxford, 1989.

[44] A. Giachetti, "Matching techniques to compute image motion," *Image and Vision Computing*, vol. 18, no. 3, pp. 247–260, 2000.

[45] J. Fernandez and V. Bhagavatula, "Partial-aliasing correlation filters," 2015.

[46] Q.-T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo, "On determining the fundamental matrix: Analysis of different methods and experimental results," 1993.

[47] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[48] B. K. Horn, "Recovering baseline and orientation from essential matrix," *J. Optical Society of America*, 1990.

[49] R. Wang and F. P. Ferrie, "Automatic registration method for mobile lidar data," *Optical Engineering*, vol. 54, no. 1, pp. 013 108–013 108, 2015.

[50] S. E. Budge and N. Badamikar, "Automatic registration of multiple texel images (fused lidar/digital imagery) for 3D image creation," pp. 873 107–873 107–10, 2013. [Online]. Available: http://dx.doi.org/10.1117/12.2016199

[51] S. E. Budge and X. Xie, "Improved registration for 3D image creation using multiple texel images and incorporating low-cost GPS/INS measurements," pp. 90 800O–90 800O–10, 2014. [Online]. Available: http://dx.doi.org/10.1117/12.2050711

[52] O. Küng, C. Strecha, A. Beyeler, J.-C. Zufferey, D. Floreano, P. Fua, and F. Gervaix, "The accuracy of automatic photogrammetric techniques on ultra-light UAV imagery," in *UAV-g 2011-Unmanned Aerial Vehicle in Geomatics*, no. EPFL-CONF-168806, 2011.

[53] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[54] A. Blake and A. Zisserman, *Visual reconstruction*. MIT press Cambridge, 1987.

[55] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*. Springer, 2008, pp. 871–889.

[56] F. Cazals and J. Giesen, "Delaunay triangulation based surface reconstruction," in *Effective Computational Geometry for Curves and Surfaces*. Springer, 2006, pp. 231–276.

[57] C. Sanderson, "Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments," http://espace.library.uq.edu.au/view/UQ:224609/armadillo_nicta_2010.pdf, 2010.

[58] "Pix4d software," https://pix4d.com/download/, accessed: 2015-05-11.

[59] "Visualsfm : A visual structure from motion system," http://ccwu.me/vsfm/, accessed: 2015-05-11.