USING AERIAL RANGE FEATURE DESCRIPTOR (ARFD) FOR LIDAR STRIP ADJUSTMENT AND MOSAICING

by

YANYAN HU

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2014

Acknowledgements

I would like to express the deepest appreciation to my supervisor, Professor Venkat Devarajan, who has supported me throughout my thesis with his patience, encouragement and immense knowledge. I have been extremely lucky to have a supervisor who cares about both my individual and professional development. Without his guidance and persistent help this thesis would have not been possible. I would also like to thank Professor Jonathan Bredow and Professor Farhad Kamangar for helping me with my course works and being my committee members.

I would like to thank the Department of Electrical Engineering at UT Arlington for providing excellent education and support to me over the last five years. It is the place where I received my bachelor degree and continued as a graduate student. It prepared me with knowledge and trainings which are beneficial to my whole career. The faculty and staff are very friendly and they made my stay here a pleasure filled experience. I would like to thank my group members at Virtual Environment Laboratory. Their academic help, knowledge and suggestions are invaluable to this research.

Finally, I would like to express my gratitude to my wife Zixi and other family members for their unconditional trust and continuous support during the entire period of my studies.

November 19, 2014

Abstract

USING AERIAL RANGE FEATURE DESCRIPTOR (ARFD) FOR LIDAR STRIP ADJUSTMENT AND MOSAICING

Yanyan Hu, M.S.

The University of Texas at Arlington, 2014

Supervising Professor: Venkat Devarajan

Light Detection and Ranging (LiDAR) has become the primary surface data extraction mapping technique. Airborne LiDAR system consists of three major components, which are laser ranging and scanning unit, Global Position System (GPS) and Inertia Measurement Unit (IMU). LiDAR data are collected in a rectangular strip-wise pattern with 10% - 30% overlapping ratio between adjacent strips. When the calibration parameters within the system components are not accurately determined, systematic errors might occur within LiDAR strips. The systematic errors lead to strip discrepancies, which further affect the quality of end-user products like Digital Elevation Map (DEM) or Digital Surface Map (DSM). In order to reduce or eliminate the discrepancies, various LiDAR strip adjustment methods have been proposed.

Aerial Range Feature Descriptor (ARFD) and a corresponding ARFD-based LiDAR strip matching algorithm have been developed to solve the strip adjustment problem for two LiDAR strips. The algorithm transforms one strip onto the other in such a way that relative discrepancies between the two LiDAR strips are minimized. The transform model is equivalent to the homography matrix relating the two strips and can be determined by taking advantage of feature-based image registration technique like Scale Invariant Feature Transformation (SIFT). However, with the presence of more than

two LiDAR strips, the strip matching problem becomes a strip mosaicing problem and a more sophisticated algorithm is required to mosaic multiple LiDAR strips.

In this thesis, a LiDAR strips mosaicing algorithm is proposed and implemented and its performance is evaluated under various conditions. It takes advantages of the ARFD-based LiDAR strip matching algorithm and extends it to multiple strips. The novelty of the proposed algorithm is that it borrows and modifies the concept of bundle adjustment from the Photogrammetry and Computer Vision community and utilizes the systematic bias model to reduce/eliminate the global strip discrepancies.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

Airborne Light Detection and Ranging (LiDAR) is a kind of topographic laser profiling and ranging system, which is used to measure continuous profiles of the terrain from aircraft. Over the last decade, the airborne LiDAR system has been proven as a cost-effective tool for the generation of surface models with dense/accurate irregular points [1]. Because of its relatively high accuracy and modest complexity, the airborne LiDAR system has been found useful in various applications recently [4]-[6]. In this chapter, background information of an Airborne LiDAR system, including the general equipment description, data acquisition and data post-processing, is presented. In addition, data biases problem and its solutions are discussed.

1.1 Airborne LiDAR System

An airborne LiDAR system consists of both a ground segment and an airborne segment. The ground segment contains multiple Global Position System (GPS) reference stations and off-line processing hardware and software. The airborne segment typically contains a laser ranging and scanning unit, a Position and Orientation System (POS) which integrates another GPS and an Inertial Measurement Unit (IMU) – all mounted on an airplane.

During a flight, the laser ranging and scanning unit emits a beam of light pulse from a mounted laser emitter and receives the returned pulse after it reflects off the target area on earth. The traveling time between emitted and received pulse is measured and converted to the line-of-sight slant range by equation (1)

$$R = \frac{c}{2} \cdot t_L \tag{1}$$

where,

R is the distance between the laser ranging and scanning unit and target area.

c is the speed of light.

t_L is the travelling time of a light pulse.

Instead of discrete pulses, It is also possible to determine the range if the laser unit emits light continuously. Such a signal is called continuous wave signal. In this configuration, the intensity of laser light has to be modulated in order to carry out ranging [7]. While the laser ranging and scanning unit measures the range, the onboard POS independently stores the carrier phase information of GPS and the orientation data of the IMU. At the same time, the on-ground GPS stations gather the necessary data at known fixed positions on earth for later off-line computing of differential GPS (DGPS) positions of the airborne platform. With the aid of DGPS and IMU, the position and orientation of the airborne platform can be determined with very high accuracy (centimeter to decimeter accuracy for position and one-hundredth degree accuracy for orientation [7]). Then the position and orientation data are post-processed with the range data and the scanning angle from the laser ranging and scanning unit to determine the 3D coordinate, which are X (latitude), Y (longitude) and Z (elevation) of the target area. The detailed-explanation of the post-processing will be provided in later sections. Besides the 3D coordinates, the amplitude of the returned pulse, marked as I, is also recorded as the intensity value of the target area.

Since the laser ranging and scanning unit emits beams of laser to the ground, ideally the target area detected by the laser beam should be a point or as small as the cross section of the laser beam. However, in reality, the laser beam is not perfectly cylindrical and therefore some divergence of the laser beam is inevitable. As shown in Figure 1-1, the beam waist is the narrowest part of the beam. From the waist, the beam diverges by an angle γ, which is known as the beam divergence angle, thus causing a

significant footprint to form on the ground. The calculation of the area of the LiDAR footprint can be found in [8]. The size of footprint varies according to the applications. Typically, the footprint diameters and footprint spacing are less than 0.5m and 1.5m respectively [9]. The sensor receives the entire return from the footprint and respectively assigns an average value as the elevation and intensity of the target area.

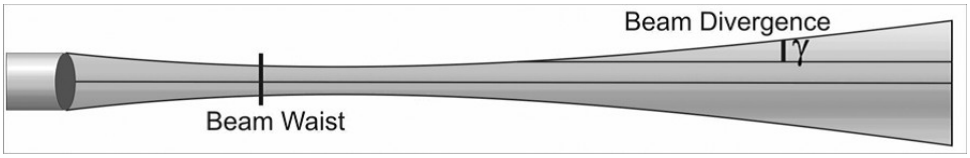


Figure 1-1 Illustration of LiDAR Beam Divergence [2]

One important fact to note here is that the LiDAR range data has multiple returns and the number of returns depends on the natural characteristic of the target area. For example, as shown in Figure 1-2, a tree may have multiple returns because each level of the tree may reflect a certain portion of the light pulse, plus that the bare ground can reflect the rest of light pulse as well. Each one of the multiple returns can be found useful in different applications. Therefore, care must be taken to consider the correct return based on the end application.

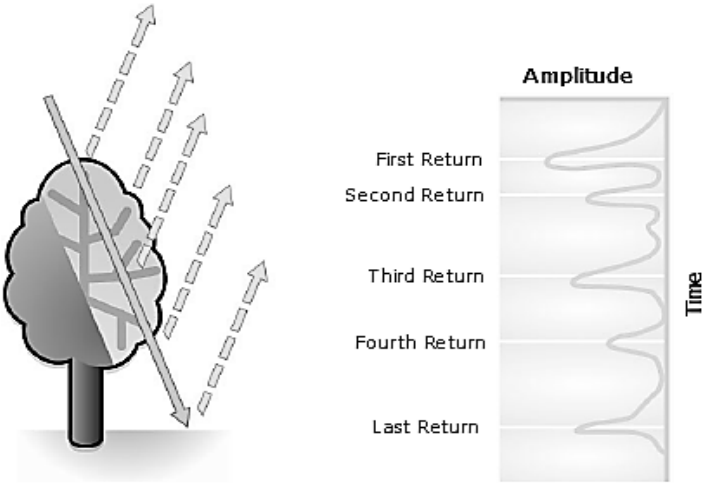


Figure 1-2 Multiple LiDAR Returns [3]

1.2 Airborne LiDAR Data Acquisition

Airborne LiDAR data collection is usually conducted in a rectangular strip-wise pattern. The rectangular strip is also called a swath. The width of each swath is determined by the flight height and the maximum scanning angle, given in equation (2).

$$SW = 2h \tan\frac{\theta}{2} \tag{2}$$

where,

SW is the width of the swath.

h is the height of the airborne platform from the ground.

θ is the scanning angle.

Within each swath, the data acquired by a specific airborne LiDAR system is determined by the following parameters.

1.2.1 Scanning Frequency

Scanning Frequency is the number of light pulses emitted by the laser unit per second [10]. Older instruments emitted a few thousand pulses per second, while modern systems are able to support frequency up to 200 KHz [9]. The scanning frequency is directly related to the density of returned light pulses, a.k.a. returns, obtained by the laser unit. Given that the flight height and speed are constant, an airborne LiDAR system operating at higher scanning frequency will generate a higher number of returns than a system operating at lower frequency. Equivalently, a high-frequency system can generate desired return density by operating on an aircraft that flies higher and faster than an aircraft carrying a lower frequency system, thereby reducing flying time and acquisition costs [10].

1.2.2 Scanning Pattern

Scanning pattern is the spatial arrangement of the light pulses emitted to and returning from a flat surface while the airborne platform is flying forward. It depends on

the movement of the laser unit across the flight line. Four scanning patterns, which are zigzag, parallel across the swath, elliptical and parallel along the swath are shown in Figure 1-3

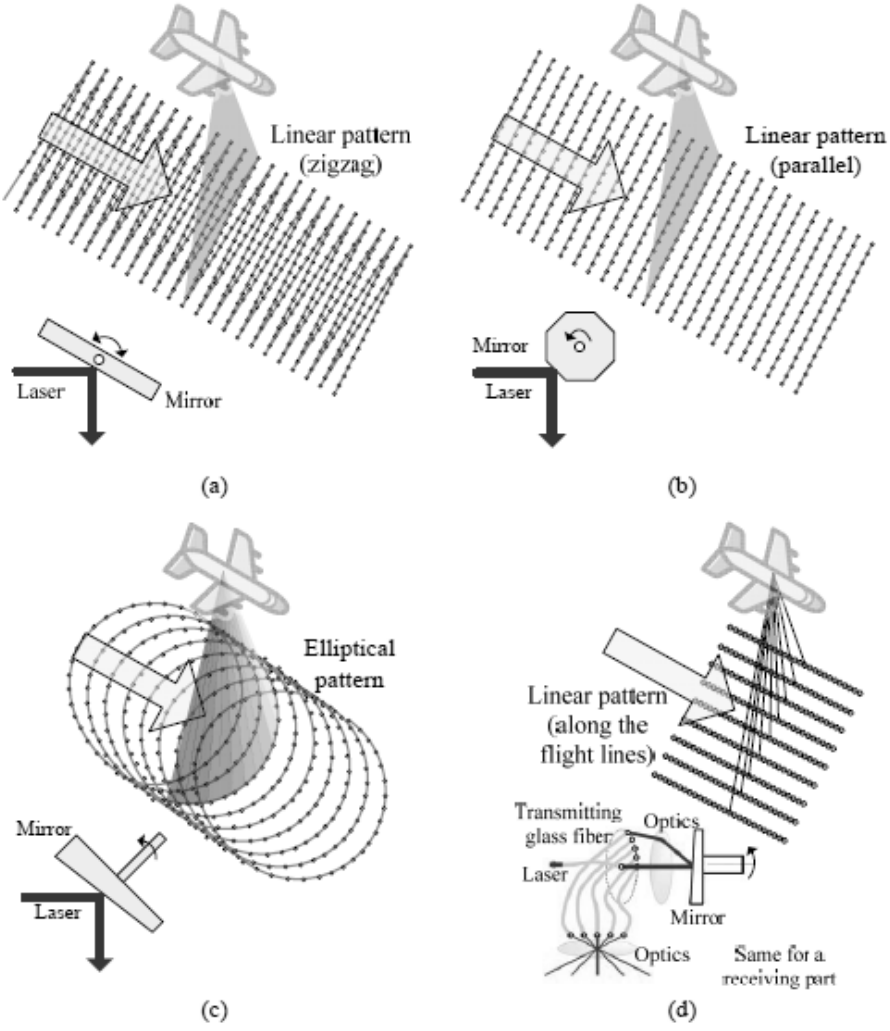


Figure 1-3 Scanning Patterns and Devices [2]

The scanning pattern is affected by the scanning device. As shown above, the zigzag pattern is generated by using an oscillating mirror. Returns are generated continuously in both scanning direction. The parallel across the swath pattern is

generated by a rotating polygon and the returns are generated in one direction of the scan only. The elliptical pattern is generated by using a nutating mirror, which moves the laser beam along an elliptical path below the aircraft. In this pattern, the ground is scanned twice from different direction. The parallel along swath pattern is generated by a fiber scanner, which consists of two arrays of glass fibers (transmitting and receiving arrays) arranged at one end in a circle and at the other end in a line [2]. Among all the scanning devices, the most popular ones in the commercial LiDAR systems are the oscillating mirror and rotating polygon, both of which are linear scanners [11-12]

1.2.3 Beam Divergence Angle

As mentioned in the previous section, the laser beam emitted by a LiDAR system deviates from the center line of beam propagation and forms a larger footprint. Because the total amount of energy per pulse remains constant, larger beam divergence angle will lead to a larger footprint area, and eventually a lower signal-to-noise ratio.

1.2.4 Scanning Angle

The scanning angle is the maximum angle of the beam which is directed away from the “focal” plane of the LiDAR instrument [10]. Typically, the scanning angle supported by most systems does not exceed 15 degree.

1.2.5 Footprint Diameter

Footprint diameter is the diameter of the intersection of a laser beam and a perpendicular plane from a distance equal to the flight height. As discussed above, the foot print diameter is affected by the beam divergence angle. With a constant beam divergence angle, higher flight height will generate a larger footprint. Moreover, as the laser unit scans to the side, the footprint becomes larger depending on the instantaneous scanning angle [29]. With all the factors considered, equation for calculating the footprint diameter is shown as follows:

$$d = \frac{H \cdot \gamma}{cos^2\theta} \quad (3)$$

where,

d is the footprint diameter.

H is the flight height

γ is the beam divergence angle and

θ is the instantaneous laser unit scanning angle.

1.2.6 Pulse Length

Pulse length describes the duration of each complete laser pulse. It is used to determine the range resolution of the pulse in multiple return systems, or the minimum distance between consecutive returns from a pulse [10]. The pulse length is measured in nanoseconds (ns). Smaller pulse length gives higher range resolution.

1.2.7 Number of Returns

As discussed previously, the LiDAR range data has multiple returns. The number of returns determines the maximum number of returns that can be recognized from a single laser beam. Most modern systems can identify up to five multiple returns.

1.2.8 Footprint Spacing

Footprint spacing is the nominal distance between the center of two consecutive footprints in the same scanning line. It is determined by the scanning frequency, the flight height and the speed of the airborne platform. The footprint spacing determines the spatial resolution of LiDAR data

1.3 LiDAR Data Post Processing

The final LiDAR point coordinates is the product of combining the range measurement with DGPS and IMU. The DGPS provides the position information of the

platform with respect to the reference and, the IMU provides the roll, pitch and yaw parameters of the platform. At each timestamp, the laser scanner picks up the returned pulse; the corresponding measurement from DGPS and IMU is also recorded. With proper calibration, the exact coordinates of the target point on the ground is determined by the following LiDAR equation [13]

$$\vec{X}_G = \vec{X}_0 + R_{yaw,pitch,roll}\vec{P}_G + R_{yaw,pitch,roll}R_{\Delta\omega,\Delta\phi,\Delta\kappa}R_{\alpha,\beta}\begin{bmatrix}0\\0\\-\rho\end{bmatrix} \quad (4)$$

where,

$\vec{X}_G$ is the position of laser point on the ground

$\vec{X}_0$ is the vector from the origin of the reference

$R_{yaw,pitch,roll}$ is the rotation matrix relating the ground and the IMU coordinate system

$\vec{P}_G$ is the lever-arm which is defined by the spatial offset between the laser unit and the IMU

$R_{\Delta\omega,\Delta\phi,\Delta\kappa}$ is the rotation matrix defined by the boresight angle $\Delta\omega, \Delta\phi, \Delta\kappa$ which are rotation offsets between the IMU and the laser unit coordinate

$R_{\alpha,\beta}$ is the rotation matrix relating the laser unit to the laser beam coordinate where α is the encoder angle along the flight path and β is the encoder angle perpendicular to the flight path, as shown in Figure 1-4..

$\vec{\rho}$ is the laser range whose magnitude is equivalent to the distance from the laser firing point to its footprint.

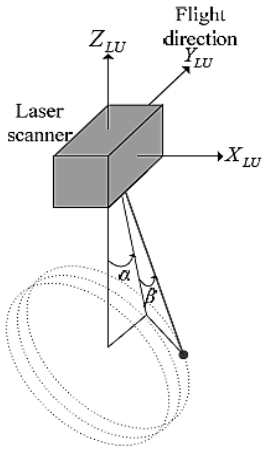


Figure 1-4 Illustration Encoder Angles [2]

Figure 1-5 illustrates the coordinate system of each component (ground, IMU, laser unit and laser beam) and the quantities involved in the LiDAR equation (4).

As mentioned in previous section, linear scanner is the most popular device in modern LiDAR systems, and therefore it is the focus of this research as well. For linear scanner, the laser beam direction can be represented by the encoder angle β only with α set to be zero.

Post processing is performed offline to obtain the ground points using the above equation. The LiDAR point cloud will be obtained after the post processing. There are multiple chances for the above setup to falter, such as the following

1. The calibration has to be up to date to have the correct parameters for the laser scanning unit and the offsets and rotations.
2. The IMU needs to have been calibrated properly to supply the correct roll, pitch and yaw parameters.

The above set of steps constitutes quality assurance. However, during flight mission, random errors can occur in properly calibrated systems too. Therefore, once the points have been obtained, quality control must be performed in order to ascertain the

accuracy of the data in hand. There is a likelihood that the system might need to be recalibrated based on the quality control steps [9].

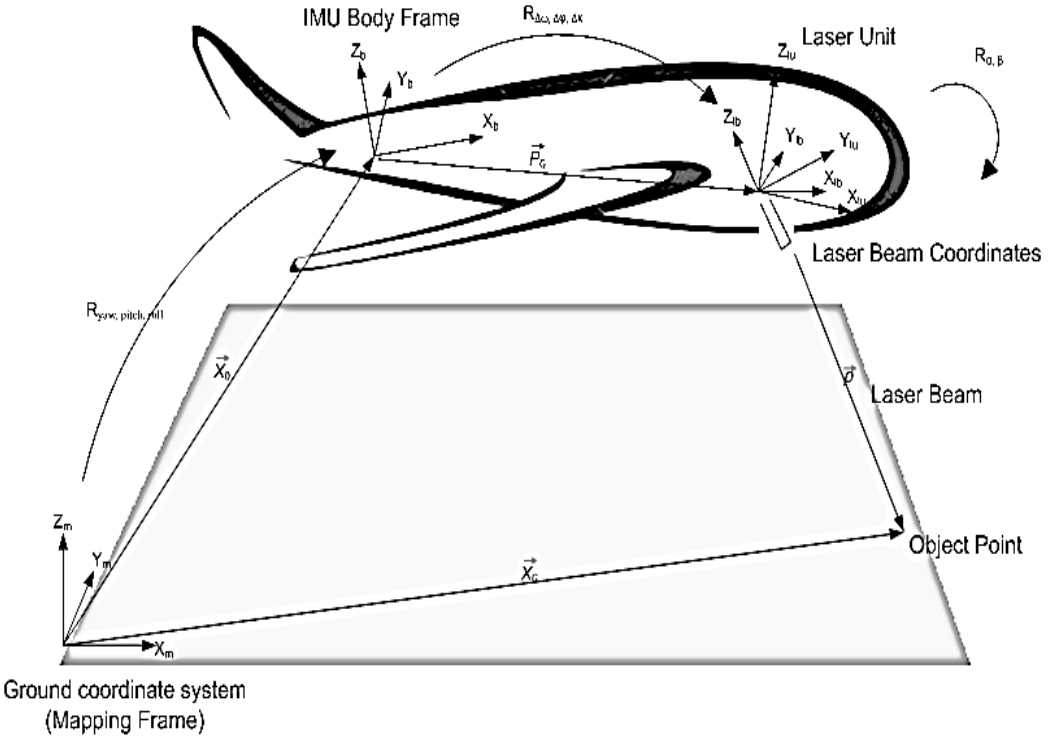


Figure 1-5 Coordinate Systems and Involved Quantities in the LiDAR Equation [14]

1.4 LiDAR Systematic Biases and Strip Adjustment

As mentioned in previous section, the LiDAR data are collected in strips. The adjacent LiDAR strips are required to have 10% - 30 % [15] overlapping area for the purpose of Quality Assurance and Quality Control (QA/QC). Ideally, there should be no visible or measurable differences between overlapping LiDAR strips, except for some random noise. However, strip differences frequently occur, due to systematic biases.

Systematic biases mainly refer to the biases in the boresight angles and lever-arm offsets relating the system components as well as the biases in the system

measurement such as encoder angles and laser ranges. The boresight angles, lever-arm, encoder angle and laser range have been defined in equation (4). The impact of these biases can be derived though mathematical analysis of the LiDAR equation and the details of the analysis will be discussed in Chapter 3. These system biases result in discrepancies between the LiDAR strip and the ground truth, where ground truth refers to the actual *X*, *Y* and *Z* values of an identifiable point.

With the purpose of reducing, or ultimately eliminating these discrepancies found in strip overlap areas, various LiDAR strip adjustment methods have been proposed in the past. For the first generation of commercial LiDAR system, strip adjustment was firstly aimed at removing only the elevation difference, because only vertical accuracy of the LiDAR data was specified [16]. As the LiDAR equipment improved, horizontal precision evaluation is now part of the process.

Theoretically, the concept of the LiDAR strip adjustment process is simple and can be divided into the following steps: firstly, the difference in the overlap area should be identified and the transformation parameters can be determined by using a suitable geometric model. Then the transformation parameters can be applied to correct/adjust the strips. However, in practice, LiDAR points are irregularly distributed, which means that the same object space is randomly sampled in the spatial domain in every strip, and therefore it is very unlikely (nearly impossible) to have point to point correspondence in the overlap areas of two strips. The lack of conjugate points makes it difficult to correctly identify the strip difference and increases the complexity of strip adjustment.

To explain the strip adjustment process mathematically, let $P\{(x_i^p, y_i^p, z_i^p),\ i = 0,1,\dots,n\}$ and $Q\{(x_j^q, y_j^q, z_j^q),\ j = 0,1,\dots,m\}$ each be a LiDAR point cloud in 3D and belong to different LiDAR strips. P and Q are the irregular spatial samples of the same area. As mentioned earlier, there is no exact conjugate points between P and Q. The task in strip

adjustment therefore is finding a transformation *T* that minimizes the relative distance between the two point clouds. This is described by the following equation [15]

$$\min_{T} \|P - T(Q)\| \tag{5}$$

The transformation $T = [T_x\ T_y\ T_z]^T$ can be established between two overlapping LiDAR strips or between a reference surface and a LiDAR strip [15]. In the first case, the transform is only applied on one data set, thus only the relative strip discrepancies between two strips is reduced. In the second case, the transformation is applied on all the strips in order to improve the absolute accuracy of the LiDAR data cloud. With the first case being accomplished in [9], the focus of this research is on the second case.

1.5 ARFD and Image Mosaicing

Aerial Range Feature Descriptor (ARFD) [9], as its name suggests, is a feature descriptor for LiDAR range features. Together with its corresponding LiDAR strip matching algorithm, ARFD is used to solve strip adjustment problem for two LiDAR strips by combining the power of LiDAR 2.5D elevation data with keypoint detector and descriptor.

The LiDAR 2.5D elevation data is obtained by rescanning an irregular grid and interpolating the data onto a regular raster grid [9]. Its advantages are 1) LiDAR data can be represented directly in the regular raster grid. 2) Digital image processing technique can be directly applied on. The keypoint detector finds interesting keypoints in the 2.5D representation of LiDAR strip by using the scale-space blob-detection technique. Elevation statistics obtained from subdivisions of the neighborhood patch surrounding a keypoint are defined as the keypoint descriptor. Each element of such a descriptor is referred to as one feature dimension and all elements together form the feature space. Once all the keypoints and descriptors are obtained from the overlapping area of two

overlapping strips, correspondences are found using the nearest neighbors of each point in the feature space. These matches can be used to find the homography that is used to transform one strip onto another [9]. Details of ARFD and the corresponding LiDAR strip matching algorithm will be introduced in Chapter 2.

Despite of its outstanding performance [9] in matching two LiDAR strips, the ARFD-based strip matching algorithm reaches to its limitation with the presence of multiple strips which partially overlap at least two other strips. The difficulty lies in the need of minimizing global discrepancies among all the LiDAR strips, while the strip matching algorithm can only eliminate the relative discrepancies between two strips.

In the image processing and computer vision communities, similar problems are tackled by many researchers and various solutions have been proposed. For example, Brown and Lowe [35] proposed an image stitching technique that mosaics several photos taken at different positions into a panoramic image. The basic idea of their approach is to obtain the camera parameters of each photo by minimizing differences of all the correspondences jointly from all the images, which is the similar to the criteria of solving the problem of matching multiple LiDAR strips. Therefore, adopting the image mosaicing concepts and techniques is a reasonable approach to extend the ARFD-based LiDAR strips algorithm to multiple strips. Detailed reasoning and speculation will be provided in Chapter 3.

1.6 Thesis Goal and Outlines

In this thesis, by exploiting the advantage of AFRD in matching LiDAR strips and the analysis of systematic biases, a method is proposed that mosaics multiple LiDAR strips which are contaminated by the system bias. The goal is to eliminate the horizontal

discrepancies among the LiDAR strips and directly/indirectly estimate the system bias parameters.

The structure of the thesis is as follows:

Chapter 2 presents an overview of the ARFD and the algorithm to match two overlap LiDAR strips.

Chapter 3 reviews the literature of image mosaicing and proposes the LiDAR strip mosaicing method based on the systematic biases model.

Chapter 4 describes the experiment carried out and analysis of result.

Chapter 5 summarizes the study of the proposed method and provides recommendation for future work.

Chapter 2

ARFD and LiDAR Strip Matching

Aerial Range Feature Descriptor (ARFD) [9] is a newly developed feature descriptor for the LiDAR range features, aiming to match overlapping LiDAR strips by feature detection and matching techniques. The range features are defined as keypoint features from the interpolated range data. The details of ARFD and the LiDAR strip Matching algorithm will be discussed in this chapter.

2.1 Introduction to ARFD

Inspired by SIFT [17], GLOH [18], and SURF [19], which are image intensity based feature descriptors and widely used in the computer vision community, ARFD is designed to be based on the elevation data of the LiDAR strip but still takes advantage of feature-based image matching technique. ARFD evaluates the statistics of elevation data in the neighborhood of the detected key point location (the key point detection algorithm will be discussed in the LiDAR strip matching algorithm). For every key point location, a 16×16 patch centered at the key point is extracted in the corresponding elevation data. Then the patch is subdivided into 16 sub-patches of size 4×4. The extracted patch and sub-patches are shown in Figure 2-1. For each sub-patch, the mean, variance, skewness and kurtosis of the 16 samples inside are calculated, which form a 4 dimensional descriptor. Following the order of top-left-to-bottom-right, every inside sub-patch generates a 4 dimensional descriptor. All 16 descriptors are concatenated to form the final 64 dimensional descriptor, which is the ARFD for the corresponding keypoint.

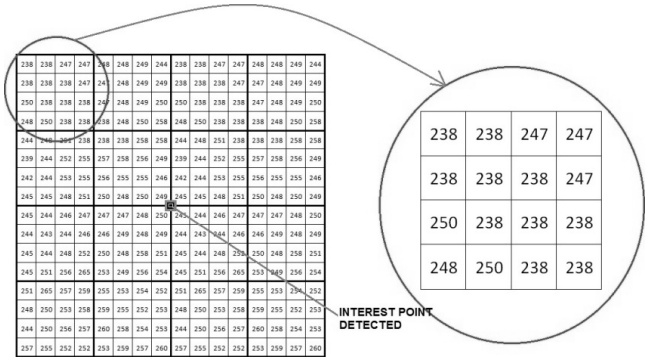


Figure 2-1 Extracted Patch and Sub-patches [9]

2.2 ARFD Based LiDAR Strip Matching Algorithm

The goal of the ARFD based LiDAR strip matching algorithm is to establish the transformation between two overlapping LiDAR strips to minimize the relative horizontal discrepancy between the strips. The algorithm can be summarized into the following steps:

1. Interpolate LiDAR data clouds into regular raster grid.
2. Automatically detect the overlap area.
3. Automatically detect keypoints in the overlap area from each LiDAR strip.
4. Generate ARFD for each keypoint
5. Find likely keypoint matches in both LiDAR strips.
6. Use RANdom Sample Consensus (RANSAC) to eliminate outliers from the likely matches and estimate the homography/projective transformation
7. Apply homography transformation on one of the strips and then align them with each other.

2.2.1 LiDAR Data Interpolation

Since the LiDAR points cloud are irregularly sampled, it is necessary to interpolate the LiDAR data of both strips into an independent regularly-spaced raster grid. By introducing the interpolation process, each LiDAR swath data creates two regular LiDAR images, which are intensity (*I* value) image and elevation (*Z* value) image. One important fact to note here is that the size of the LiDAR image not only depends on the area covered by the LiDAR strip, but also on the arbitrarily selected resolution, which is defined as the actual area represented by each pixel. The interpolation technique employed in the strip matching algorithm is the Inverse Distance Weighting (IDW) interpolation algorithm. Firstly, after the grid is set, the intensity and elevation data of each point is assigned to the nearest cell as well as its closest 8 neighbor cells, as shown in Figure 2-2. After the first round of assignment, in the next round, each empty cell is identified and it “borrows” the data from its non-empty 8 neighbor cells, with the corresponding Euclidean distance recorded. The “borrowing” process will repeat 3 times and the Euclidean distance will be updated. Once the iteration is finished, most of the cells will be assigned some points, and then the elevation and intensity value in each cell is determined as a weighted average of the data of the points assigned to the cell, with the weights being a function of the inverse Euclidean distance between the current cell and the cells from which the current cell “borrows” data. The functions for calculating the interpolated elevation and intensity values are shown in equation (6) and (7), respectively.

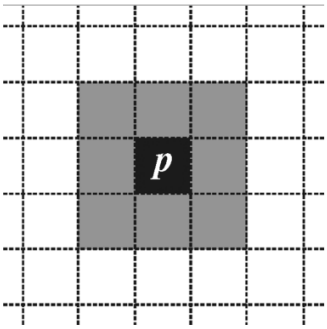


Figure 2-2 Adaption of Inverse Distance Weighting Interpolation Algorithm [9]

$$E_{x,y} = \frac{\sum_{k=1}^{n} \frac{e_k}{d_k}}{\sum_{k=1}^{n} \frac{1}{d_k}} \tag{6}$$

$$I_{x,y} = \frac{\sum_{k=1}^{n} \frac{i_k}{d_k}}{\sum_{k=1}^{n} \frac{1}{d_k}} \tag{7}$$

where,

$E_{x,y}$ and $I_{x,y}$ are the elevation and intensity at current cell location (x, y).

e_k and i_k are the elevation and intensity values of the individual data points that have been assigned to the cell location.

n is the number of cells from which data is lent

d_k is the distance between the current cell and the cells from which the current cell borrows data

2.2.2 Automatic Overlap Area Detection

The overlap area between two LiDAR strips is detected automatically by comparing boundaries of both strips. For two horizontal parallel strips configuration, as shown in Figure 2-3 , the shape of the overlap area, is rectangle. The top boundary of the

overlap area is the maximum value of both the strips' top boundaries and similarly, the bottom boundary is the minimum value of both the strips' bottom boundaries. The left and right boundaries are also determined in the similar way.

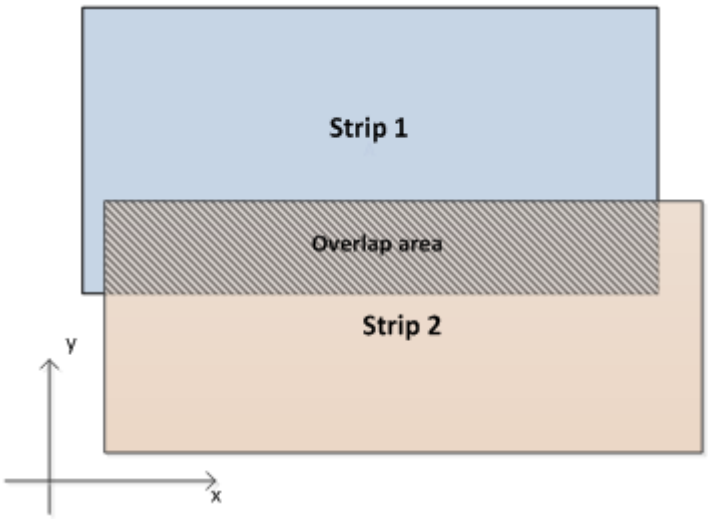


Figure 2-3 Strip Overlap Configuration

2.2.3 Automatic Keypoint Detection

After determining the overlap area, keypoints are extracted from the interpolated LiDAR elevation images. In the LiDAR strip matching algorithm, the scale-space blob detection is selected and proved as "best suited for extracting the most interesting features (a.k.a. keypoints) for LiDAR data matching" [9]. The scale-space blob detection is generalized as following.

Firstly, a pyramid of image is constructed for multiple octaves, as shown in Figure 2-4. Each Octave contains multiple LiDAR images of the same scale, starting with 2 times the initial size of the image and shrinking by a factor of 2 each time. A total of 4 such octaves are created for the purpose of covering multiple scaling variations between images being matched.

After construing the image pyramid, a series of 2D Gaussian filters with different variance are applied to the octaves to obtain various blurring levels. This is followed by the subtraction between the adjacent images within the same octave to obtain the Difference of Gaussian (DoG) image (see Figure 2-4). The DoG is an approximation of the Laplacian of Gaussian (LoG) multiplied by a constant factor [17] and the LoG is used to extract the scale-space blobs.

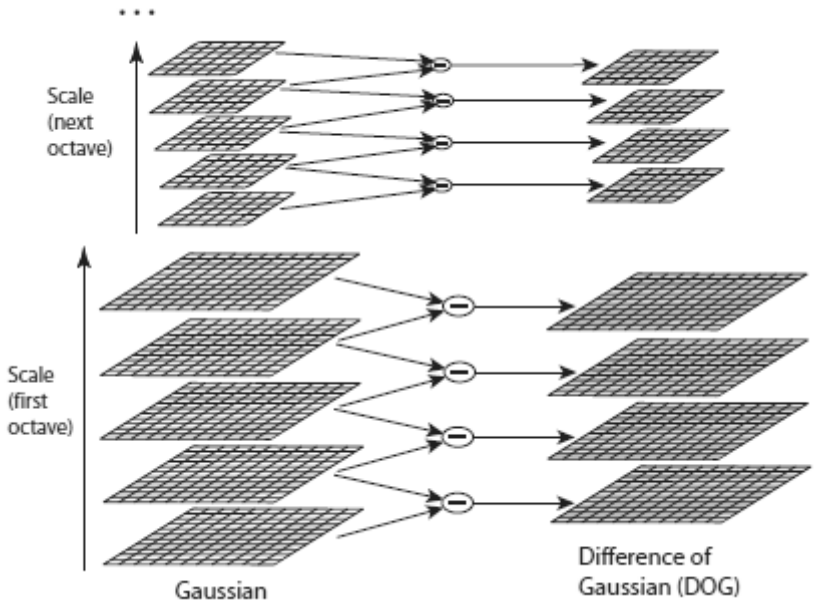


Figure 2-4 Image Pyramid and DoG Image [17]

After the DoG images are obtained, a 3D non-maxima suppression is performed. For each sample in the DoG images, its 26 3D neighbor samples (8 in the current scale and 18 in the two adjacent scales) are considered, as shown in Figure 2-5. The samples which are the extremes among the 26 neighbors and exceed a threshold are selected as the strong keypoints. Finally, after removing the points on the edges, the surviving points are the scale-space blob keypoints.

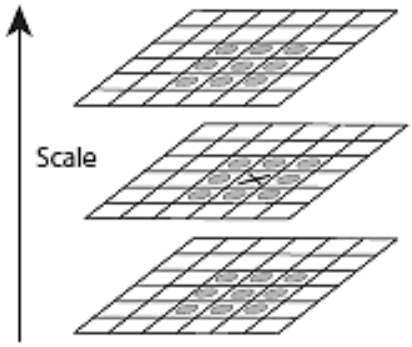


Figure 2-5 3D Neighbor Samples [17]

2.2.4 Descriptor Generation

For every keypoint detected in the previous step, a corresponding ARFD is generated. Details of ARFD have been introduced in the previous section. In addition, in the LiDAR strip matching algorithm, an adaptive technique is also adopted. The descriptor is selected for each keypoint in an adaptive fashion by choosing to use the elevation or intensity data based on a higher kurtosis value of either of the modalities [9]. It turns out that this adaptive technique helps improve the final LiDAR strip matching performance.

2.2.5 Keypoints Matching Algorithm

The keypoints matching method adopted in the LiDAR strip matching algorithm is based on the Nearest-Neighbor Distance Ratio (NNDR).

1. For each keypoint and its corresponding descriptor in the first strip, the Euclidean distances in descriptor space between this point and all the points in the second strip are calculated.
2. The NNDR of the current keypoint in the first strip is the ratio of the smallest distance over the second smallest distance.

3. If the ratio is greater than the threshold (normally in the range of 1.2 to 1.5), the keypoint corresponding to the smallest distance is picked as the best match. Otherwise, no best match is selected for the current key point.

In the end, all of the best matches are identified as positive matches and the rest are identified as negative matches.

2.2.6 Removing Outlier and Homography Estimation by Using RANSAC

RANSAC [20] is a general parameter estimation approach designed to cope with a large proportion of outliers in the input data [21]. In the LiDAR strip matching algorithm, the RANSAC is adopted to eliminate the outliers of keypoint correspondences from the positive matches and estimate the homography matrix. The algorithm of RANSAC is generalized as following [21]:

1. Select randomly the minimum number of points required to determine the model parameters.
2. Solve for the parameters of the model.
3. Determine how many points from the set of all points fit with a predefined tolerance ϵ.
4. If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ, re-estimate the model parameters using all the identified inliers and terminate
5. Otherwise, repeat steps1 through 4 (maximum of *N* times).

After the outliers are eliminated by the RANSAC algorithm, the homography matrix is estimated again by using all the inliers.

The homography between a pair of corresponding keypoints is defined as:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (8)$$

$$\lambda \vec{X}' = H\vec{X} \quad (9)$$

where $\vec{X}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ and $\overrightarrow{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ are two keypoints in homogenous form, H is the 3×3 homography matrix and, λ is a non-zero constant.

Dividing the first and second row of equation (8) by the third row leads to the following equations:

$$-h_1 x - h_2 y - h_3 + (h_7 x + h_8 y + h_9) x' = 0 \quad (10)$$

$$-h_4 x - h_5 y - h_6 + (h_7 x + h_8 y + h_9) y' = 0 \quad (11)$$

then the two equations above can be written in matrix form as:

$$A_i \vec{h} = \mathbf{0} \quad (12)$$

Where,

$$A_i = \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{bmatrix}$$

$$h = [h_1 \quad h_2 \quad h_3 \quad h_4 \quad h_5 \quad h_6 \quad h_7 \quad h_8 \quad h_9]$$

The homography matrix contains 8 degrees of freedom. Since each keypoint correspondence provides 2 equations, at least 4 pairs of keypoint correspondences are required. Finally the least square problem of equation (12) can be solved by finding the $\vec{h}$ that minimizes $\|Ah\|^2$. This method is called the Direct Linear Transform (DLT) [43] algorithm for homography calculation.

2.2.7 Homography Application and Strip Alignment

After the homography matrix is determined, the transformation is applied on one of the LiDAR strips. Then the two LiDAR strips are aligned with each other to present the

final matching result, an example of which is shown in Figure 2-6. The green square in the middle highlights the detected overlapping area.

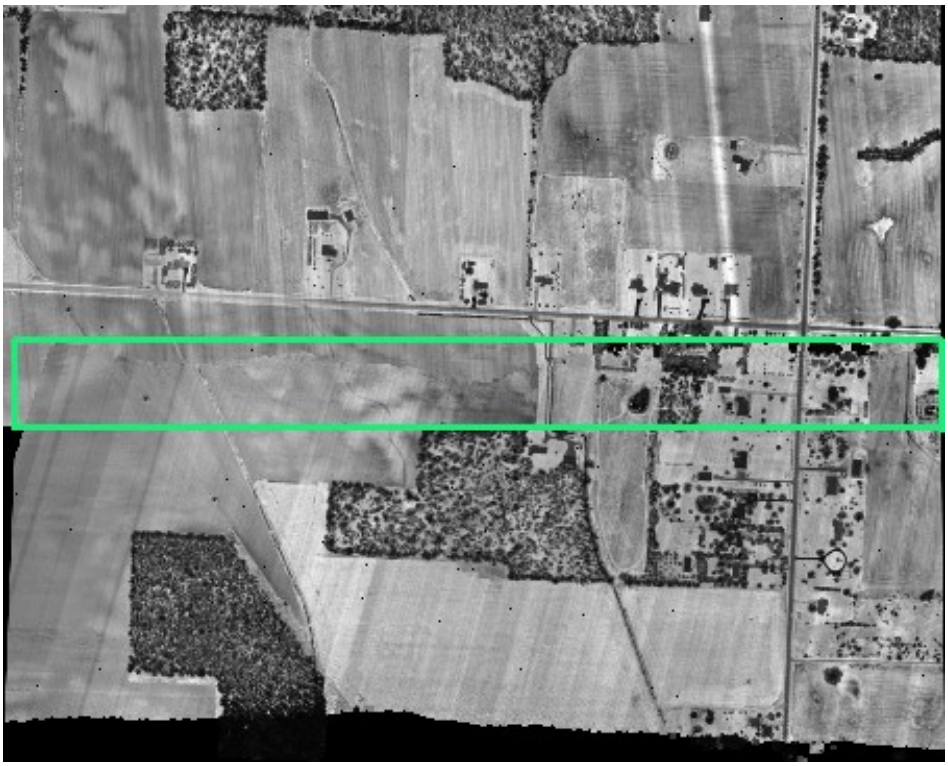


Figure 2-6 LiDAR Strip Matching Final Result [9]

Chapter 3

ARFD-based LiDAR Strip Mosaicing Algorithm

The result of the ARFD-based LiDAR strip matching algorithm in the previous chapter showed a successful alignment of two overlapping LiDAR strips. Naturally, one reasonable question could then arise: will this algorithm work if there are more than two LiDAR overlapping strips? This is an especially significant question to answer since, in practice, due to the limitations of flight height and scanning angle, two LiDAR strips are far from enough to cover the whole target region or a region of interest. Multiple LiDAR overlapping strips are always required. Besides, as mentioned previously, the LiDAR strip-matching algorithm presented in Chapter 2 merely minimizes the *relative* horizontal discrepancies between the two LiDAR strips. In other words, there is an assumption that one of the two strips is considered as completely correct in terms of its representation of the ground truth. The assumption could be valid for only two strips, because the biases within each strip are very small and can be ignored.

At this point in the discussion, we want to make a distinction between photo-based photogrammetry and LiDAR based Lidargrammetry. In photogrammetry, especially before the arrival of differential GPS, two photos matched using key point matching approaches never really got very close in terms of representing the ground truth. This is because, there were no inherent Z values collected using photos. Errors associated with the generation of Z values from photos were caused by many factors: scanning errors, internal camera calibration errors, lack of accurate orientation information for the airplane that took the pictures etc. The last mentioned issue was effectively removed after differential GPS systems came in to vogue. Nevertheless, the generation of Z values from photos still needed matching overlapping regions of a pair of photos and calculating heights of ground points in that overlapping area using parallax information. Even then,

additional ground control points were needed to anchor the whole model to the real world (earth).

The situation with LiDAR is different for many reasons. First of all, the LiDAR system calculates height information from measurement of a laser unit. Thus the outputs of the system as mentioned before are the X, Y and most importantly Z, in addition to the intensity. The main error left is the calibration errors or what we refer to as “biases”. This is true even when we use just two strips and match them using key point matching techniques. Although such a match is far more accurate and connected to the real earth than would a photo pair be, when multiple strips are aligned together, the ignored small biases will propagate along each strip and finally could lead to significant errors. Therefore, multiple LiDAR strips matching problem, or more accurately, the *LiDAR strip mosaicing problem,* cannot be solved by directly applying the strip-matching algorithm on the consecutive pairs.

The difficulty in the LiDAR strip mosaicing problem is similar to the one existing in the digital image mosaicing problem. When creating mosaicing of digital images, concatenation of pairwise homographies would cause accumulated errors and disregard multiple constraints among images. In the computer vision community, adopting the bundle adjustment technique to estimate the camera parameters jointly solves this problem. However, the bundle adjustment method could not be directly applied to the LiDAR strips because the proper “camera model” for LiDAR strip does not exist. Therefore, it is necessary to develop a LiDAR version of bundle adjustment technique in order to solve the LiDAR strip mosaicing problem.

In this chapter, because of the similarity between LiDAR strip mosaicing and the digital image mosaicing, the literature of image mosaicing methods and bundle adjustment is briefly reviewed. Then the mathematical model of LiDAR systematic biases

is introduced. Finally, a LiDAR strip mosaicing algorithm, which is based on the systematic biases model and takes advantages of the ARFD, is proposed.

3.1 Brief Review of Image Mosaicing Methods

Image mosaicing is a synthetic process, which generates a composition from a sequence of images by understanding geometric relationships among them [22]. Mosaicing of images has been in practice for a long time; even before the digital image technology came about. For example, in the photogrammetry community, the aerial mosaic was widely used to show the relative horizontal positions of terrain features. It is a series of overlapping vertical or near vertical photographs assembled in sequence on a mounting board to form one continuous picture of the terrain [23]. During that time, the mosaic is generated by manually laying down the succeeding photographs according to some index on a piece of plywood or fiberboard and then stapling them into place. With the development of computer technology and the adoption of digital image, automatic image mosaicing has been found to be important in many fields, including remote sensing, virtual reality, visual systems for flight simulation, mission rehearsal, video and image processing, and computer vision. A variety of image mosaicing methods have been proposed in the last two decades. In general, various steps needed in image mosaicing methods are image registration, stitching and blending.

3.1.1 Image Registration

Image registration refers to the geometric alignment of a set of images. The set may consist of two or more digital image taken of a single scene at different times, from different sensors, or from different viewpoints. The goal of registration is to establish geometric correspondence between the images so that they may be transformed, compared and analyzed in a common reference frame [22]. In general, the image

registration methods could be classified into frequency domain method and spatial domain method.

3.1.1.1 Frequency Domain Registration Method

The frequency domain image registration method prevails in the 1990's but now only being found in limited applications after decades. One of the most famous frequency domain registration methods is proposed by Reddy and Chatter [25]. It is based on the Fast Fourier Transform (FFT-based) and the Fourier translation rotation and scale properties, which are shown in the following equations:

$$f_1(x,y) \leftrightarrow F_1(m,n) \tag{13}$$

$$\begin{matrix} f_2(x,y) = f_1(xcos\theta_0 + ysin\theta_0 - x_0, -xsin\theta_0 + ycos\theta_0 - y_0) \\ , \quad \leftrightarrow \\ F_2(m,n) = e^{-j2\pi(mx_0+ny_0)} \cdot F_1(mcos\theta_0 + nsin\theta_0, -msin\theta_0 + ncos\theta_0) \end{matrix} \tag{14}$$

$$f_2(x,y) = f_1(ax,by) \leftrightarrow F_2(m,n) = \frac{1}{|ab|} \cdot F_1(\frac{m}{a},\frac{n}{b}) \tag{15}$$

where

$f_1(x,y)$, $f_2(x,y)$ represent images in the spatial domain. $f_2(x,y)$ is equal to $f_1(x,y)$ after translation, rotation and scaling transform

$F_1(m,n)$, $F_2(m,n)$ represent images in the Fourier domain

$\leftrightarrow$ represents the 2D Fourier transform correspondence. Expression on the left side is in spatial domain and expression on the right side is in Fourier domain.

θ_0 is the rotation parameter

x_0, $y_0$are the translation parameters and

a, b are the scaling parameters.

By using the polar coordinate transformation and cross-power spectrum, this method managed to determine the value of translation, rotation and scaling parameters.

In this method, the phase correlation technique is simple and accurate, but significant overlap ratio between images is required [25].

3.1.1.2 Spatial Domain Registration Method

Spatial domain registration methods can be either non-feature-based or feature based. The non-feature-based method is also called the direct method [31], where pixel values are used directly for matching purposes. For example, cross-correlation [26][31] is the basic approach for registration. One of the images is called a template and the other target image or simply “image”. By calculating the normalized cross-correlation coefficient between the image and templates with various incremental translations and rotations of the template with respect to the image, one can get a series of measures of the degree of similarity between the two. The template, which gives the highest value of cross correlation, is the desired match. This method is useful for images, which are misaligned only by a small rigid or affine transforms.

In the feature-based methods, features, also called keypoints, are the point representatives, which represent the salient and distinctive objects (close-boundary regions, edges, contours, line intersections, corners, etc.) that are detected automatically from the image [30]. Distinctive features are detected from the overlapping part of images (to be matched) to establish global or pair-wise correspondences. Schmid *et al* [32] evaluated several low-level feature detectors including Harris [33], improved version of Harris (ImpHarris) [32], Cottier [44], Horaud [41], Heitger [27] and Förstner [28]. The evaluation criteria used in this research are repeatability rate and information content, which represent the stability and distinctiveness of key points, respectively. The repeatability rate is defined as the number of points repeated between two images with respect to the total number of detected points, while the information is the entropy measurement of the post-processed descriptor formed by combinations of the first and

second derivatives of local pixel value. The result shows that with presence of image rotation, scale change, illumination variation, viewpoint change and camera nose, the Harris detector and its improved version have both higher repeatability rate and information content.

After detecting the feature, matching is required to determine which features come from corresponding locations in different images. The matching process can be accomplished by means of the image intensity values in their close neighborhood, the feature spatial distribution, or the feature symbolic descriptions. The feature descriptor is a kind of feature point representation, which can be invariant to different orientations and/or scales, thus is widely used as an auxiliary step in matching. Mikolajczyk and Schmid [18] reviewed and evaluated multiple descriptors including SIFT, GLOH, PCA-SIFT, and etc. The evaluation criterion is based on the number of correct matches and the number of false matches obtained for an image pair, under various conditions including affine transformations, scale changes, rotations, blur, jpeg compression, and illumination change. Among all the local descriptors, they found that David Lowe's Scale Invariant Feature Transform (SIFT) generally performs the best. Details of SIFT implementation can be found in [17].

By using the descriptors, the simplest way to find all corresponding points in an image is to compare all the features in one image against all the features in the other. However, the computational cost will be $O(n^2)$ if both of the images contain n feature points. To reduce the computation complexity, the SIFT algorithm adopts the Best Bin First (BBF) algorithm [34] to speed up the matching process. The BBF algorithm is based on the k-d tree algorithm, while the bins in feature space are searched in the order of their closest distance from the query location [17] and the search process stops after the

first 200 nearest neighbor candidates. Besides, utilizing the NNDR, discussed in the previous chapter as a comparison criterion, also speeds up the matching process.

Feature point correspondences are extracted after applying the selected matching technique. However, the matched correspondences contain both inliers and outliers. Inliers can be explained by a model with a particular set of parameter values, while outliers do not fit that model in any circumstance. To remove the outliers, RANSAC, discussed in Chapter 2, and its more sophisticated variations like PROgressive Sample Consensus (PROSAC) [42] are widely used.

The transformation matrix between images can be automatically calculated during the RANSAC process. To get a more precise result, the inliers can be used to calculate the homography matrix again after RANSAC, The major difference among various homography calculation algorithms is the choice of cost function to be minimized. As discussed in Chapter 2, the cost function in the DLT method is the algebraic distance between all the correspondences. Other cost functions like geometric distance or its approximation, Sampson error [43] can be used. Besides, techniques like data normalization and adopting nonlinear least square algorithm can further improve the result.

3.1.2 Image Stitching

Given an image set ***I***, which contains N images $\{I_1, I_2, I_3, \cdots I_N\}$ with a partial overlap at least two images. The outputs from image registration are the set of matrices ***H*** that contains all the homography matrices $\{H_{ij} \mid 1 \leq i \leq j \leq N,\ I_i \text{ partially overlaps } I_j\}$, between every pair of overlapping images and each H_{ij} will satisfy equation (16),

$$\tilde{\boldsymbol{u}}_{\boldsymbol{i}} = H_{ij}\, \tilde{\boldsymbol{u}}_{\boldsymbol{j}} \tag{16}$$

where $\tilde{\boldsymbol{u}}_{\boldsymbol{i}} = [x_i, y_i, 1]$ and $\tilde{\boldsymbol{u}}_{\boldsymbol{j}} = [x_j, y_j, 1]$ are the homogeneous image position of image I_i and I_j.

The next step is to stitch the images together to create a larger mosaic. Based on whether the common reference is directly given or not, the image stitching can be classified as two cases.

3.1.2.1 Directly Sequential Stitching

This case takes place when one image of ***I*** is considered as the common reference or the homography matrix relating the common reference with one or more images of ***I*** is known. In this case, the homography matrix between the common reference and each image of ***I*** can be easily calculated by chaining the pair-wise homography matrices. New images after proper transformation can be directly aligned with the previously composited mosaic.

Using the same notation as in the beginning of this section, let H_{kl} be the corresponding homography matrix from the image I_l to the best adjacent image I_k and H_k be the corresponding matrix from the image I_k to the common reference plan. If the common reference plane is I_k itself, the H_k becomes a diagonal matrix $I_{3\times3}$. Otherwise H_k is just an ordinary projective transformation matrix. Then, the H_l that is the corresponding homography matrix from the image I_l to the common reference plane can be calculated by equation (17)

$$H_l = H_k H_{kl} \tag{17}$$

In the case that the I_k does not directly overlap I_l, the H_l can be calculated by considering all the intermediate images $(I_m, I_n, \cdots I_p)$ that connect I_k and I_l, as is given in equation (18)

$$H_l = H_k H_{km} H_{mn} \cdots H_{pl} \tag{18}$$

After calculating the all the homography matrices H_i ($1 \leq i \leq N$), each image I_i is warped and resampled according to H_i .to generated a converted image $\hat{I}_i$, which is related to I_i by the equation (19)

$$\widehat{\boldsymbol{u}}_{\boldsymbol{i}} = H_i \, \widetilde{\boldsymbol{u}}_{\boldsymbol{i}} \tag{19}$$

Finally all the converted images can be directly stitched on the common reference for further processing. Figure 3-1 shows the 3 images (a) (b) and (c) before mosaicing. During the registration step, the homography matrices H_{ab} and H_{ac} are calculated and H_a is known as well. Figure 3-2 demonstrates result of stitching (plus further blending) all the images on the common reference (the black background).

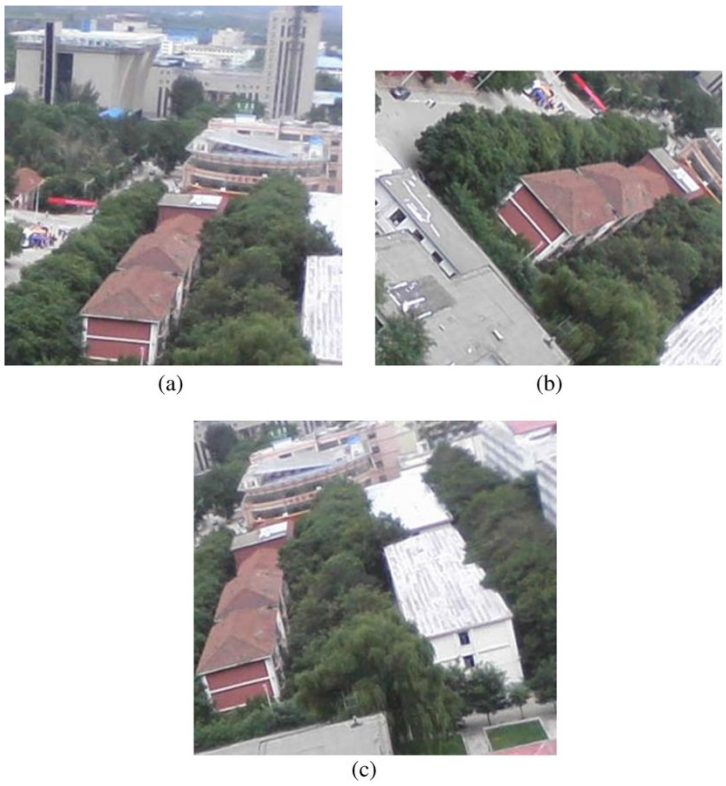


Figure 3-1 Images before Mosaicing [24]

Figure 3-2 Mosaicing Result [24]

3.1.2.2 Bundle Adjustment Based Stitching

Bundle adjustment is often used when a large number of images need to be stitched together. In this case, the common reference is not directly available. One could argue that a possible solution is to arbitrarily choose one image from the set as the common reference and then directly apply the sequential stitching, as described in the previous section for three images. Unfortunately, in this situation, the sequential stitching will suffer from the problem of accumulated misregistration errors over the large number of images. The problem is particularly severe for panoramic mosaics, where a visible gap (or overlap) will exit between the first and last images in a sequence [31] or the ends of a panoramic will join up [35].

To solve this problem, algorithms [35]-[37] based on bundle adjustment are proposed in the computer vision community. Bundle adjustment [38] was originally developed as a technique, which refines a visual reconstruction to produce jointly optimal 3D structure and viewing parameters (like the camera pose and/or calibrations). The

word ‘bundle’ refers to the bundle of light rays leaving each 3D feature and converging on each camera center. A camera projection model is adopted. The goal of implementing bundle adjustment is to minimize the global alignment error.

Let $\widetilde{\boldsymbol{X}}_{\boldsymbol{i}} = [X_i, Y_i, Z_i]$ be the *i-th* 3D feature point. Then the projection of the 3D point $\widetilde{\boldsymbol{X}}_{\boldsymbol{i}}$ onto a point $\widetilde{\boldsymbol{x}}_{\boldsymbol{ij}} = [x_{ij}, y_{ij}]$ in the image I_j is (assuming central projection):

$$\widetilde{\boldsymbol{x}}_{\boldsymbol{ij}} \sim K_j\, R_j\, \widetilde{\boldsymbol{X}}_{\boldsymbol{i}} \quad \text{and} \quad \widetilde{\boldsymbol{X}}_{\boldsymbol{i}} \sim {R_j}^{-1} {K_j}^{-1}\, \widetilde{\boldsymbol{x}}_{\boldsymbol{ij}} \tag{20}$$

Or similarly,

$$\begin{bmatrix} x_{ij} \\ y_{ij} \\ 1 \end{bmatrix} \sim \begin{bmatrix} f_j/Z_i & 0 & 0 \\ 0 & f_j/Z_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & R_{iXY} & R_{iXZ} \\ R_{iYX} & 1 & R_{iYz} \\ R_{iZX} & R_{iZY} & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \tag{21}$$

where,

$K_j = \begin{bmatrix} f_j/Z_i & 0 & 0 \\ 0 & f_j/Z_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is a calibration matrix containing the camera focal length f_j.

$R_j = \begin{bmatrix} 1 & R_{iXY} & R_{iXZ} \\ R_{iYX} & 1 & R_{iYz} \\ R_{iZX} & R_{iZY} & 1 \end{bmatrix}$ is the rotation matrix describing the rotation of the camera.

$\sim$ denotes the meaning of being proportional to.

Using equation (20), the projection of $\widetilde{\boldsymbol{X}}_{\boldsymbol{i}}$ onto another point $\widetilde{\boldsymbol{x}}_{\boldsymbol{ik}}$ in image I_k can be written as:

$$\widetilde{\boldsymbol{x}}_{\boldsymbol{ik}} \sim K_k\, R_k\, \widetilde{\boldsymbol{X}}_{\boldsymbol{i}} = K_k\, R_k\, {R_j}^{-1} {K_j}^{-1} \widetilde{\boldsymbol{x}}_{\boldsymbol{ij}} = H_{kj}\, \widetilde{\boldsymbol{x}}_{\boldsymbol{ij}} \tag{22}$$

and therefore,

$$H_{kj} = K_k\, R_k\, {R_j}^{-1} {K_j}^{-1} \tag{23}$$

where, as mentioned earlier, H_{kj} is the homography matrix from image I_j to image I_j.

Given an initial set of $\{(K_j, R_j)\}$, the rest of the camera parameters can be estimated by minimizing the following energy E through nonlinear iterative minimization technique like Levenberg-marquardt (LM) and Gauss-Newton methods [39], if the true bundle adjustment is used [31]:

$$E = \sum_i \sum_j c_{ij} \; ||\tilde{\boldsymbol{x}}_{\boldsymbol{ij}}(\tilde{\boldsymbol{X}}_{\boldsymbol{i}};\; R_j, K_j) - \hat{\boldsymbol{x}}_{\boldsymbol{ij}}\;||^2 \tag{24}$$

where, $\tilde{\boldsymbol{x}}_{\boldsymbol{ij}}(\tilde{\boldsymbol{X}}_{\boldsymbol{i}};\; R_j, K_j)$ is the predict location of the *i-th* feature in image I_j given by equation (20); $\hat{\boldsymbol{x}}_{\boldsymbol{ij}}$ is the observed location; and c_{ij} is the weight parameter.

However, the disadvantage of full bundle adjustment is that there are more variables to solve for each iteration and the overall convergence may be slower. In practice, equation (25) is suggested for simplicity and speed:

$$E = \sum_i \sum_{jk} c_{ij} \; c_{ik} \; ||\tilde{\boldsymbol{X}}_{\boldsymbol{i}}(\hat{\boldsymbol{x}}_{\boldsymbol{ij}};\; R_j, K_j) - \tilde{\boldsymbol{X}}_{\boldsymbol{i}}(\hat{\boldsymbol{x}}_{\boldsymbol{ik}};\; R_k, K_k)\;||^2 \tag{25}$$

where $\tilde{\boldsymbol{X}}_{\boldsymbol{i}}(\hat{\boldsymbol{x}}_{\boldsymbol{ij}};\; R_j, K_j)$ and $\tilde{\boldsymbol{X}}_{\boldsymbol{i}}(\hat{\boldsymbol{x}}_{\boldsymbol{ik}};\; R_k, K_k)$ are given by the second half of equation (20). Again, the non-linear least square minimization methods like LM or Gauss-Newton can be applied to get the updated camera parameters for each image.

3.1.3 Image Blending

Because of a number of unmodelled effects, such as vignetting (intensity decreases towards the edges of the image), parallax effects due to unwanted motion of the optical center and misregistration errors due to mismodelling [35], image blending is important to create the satisfactory final image mosaic. However, since the goal of the research is to eliminate the horizontal discrepancies between multiple LiDAR strips, image blending is irrelevant to what has been proposed. Therefore, no further discussion will be provided here.

3.2 LiDAR Systematic Biases Model

As discussed in the beginning of Chapter 3, despite of the fact that bundle adjustment is a very powerful tool for decreasing the global alignment error, it is not entirely applicable to LiDAR strip mosaicing problem due to the following reasons.

1. The bundle adjustment approach is based on the pinhole camera projection model. However, for the LiDAR strip and the interpolated image, the camera model is not applicable.
2. Although an initial set of $\{(K_j, R_j)\}$ is required for the nonlinear least square iterative minimization method to initiate, indicating the image I_j is the initial common reference, the final common reference for image stitching will not necessarily be image I_j . In fact, the final common reference will be determined implicitly by all the camera parameters for each image and these camera parameters are updated after every iteration in order to minimize the global alignment error. However, in the LiDAR strip, if there is no bias, all the LiDAR points will be registered perfectly within the common reference, which is the ground coordinate system preset by the ground segment of the airborne LiDAR system. Since the common reference is already given indirectly, if the bundle adjustment technique is applied, the updated final common reference is not guaranteed to converge to the ground reference system. The discrepancies between them will propagate over the whole mosaic.

Although the bundle adjustment cannot be directly applied on the LiDAR strip mosaicing, the concept of reducing global errors and decomposing the homography is still very helpful. In the literature of LiDAR strip adjustment, some the systematic biases

models were proposed and they relate the biased data with the true data, which can be viewed as a kind of 'camera model' for LiDAR.

As mentioned earlier, systematic biases exist in both system measurements and system parameters. This research is focused only on the biases in the LiDAR system parameters, which are the bore sight angles and the lever-arm offset. Let $\Delta\omega, \Delta\varphi$ and Δk be the pitch, roll and yaw boresight angles and $\Delta X, \Delta Y$ and ΔZ be the lever-arm offset along X, Y and Z axis. The directions of X, Y and Z axes will be defined later in this section. Besides, in the research, only the LiDAR point cloud containing merely the 3D coordinate and intensity of the points is used because the raw LiDAR, which also contains additional information like the laser range measurement and trajectory position of the flying platform and time-tags, is not always available to the end users.

Habib *et al* [13] proposed a 'simplified method' which only utilizes the LiDAR point cloud for strip adjustment. In their proposed method, the mathematical model of the systematic biases is created under the following conditions:

1. The method only deals with linear scanner with mirror scanning in the across-flight direction
2. Flying direction are parallel to the positive and negative directions of the Y-axis of the ground coordinate system
3. The flight lines is straight with constant attitude
4. The onboard LiDAR system is almost vertical (*i.e.* $R_{yaw,pitch,roll} \approx I_{3\times3}$ if the system flying along the positive direction of the Y-axis)
5. The LiDAR system has relatively small boresight angles
6. The mapped area is comprised of a relatively flat terrain, where the height variation are much smaller than the flying height above ground

7. The Y-axis of the ground coordinate system is defined half-way between the overlapping strips at the ground level.
8. The convention used for the laser scanner and IMU local coordinate is right forward-up (right-handed)

This systematic biases model forms the basis for the LiDAR strip mosaicing algorithm proposed in the thesis and it will be introduced in the following section.

Under the conditions above, the LiDAR equation (4) can be rewritten as:

$$\begin{aligned}\vec{X}_G &\approx \vec{X}_0 + \begin{bmatrix}\Delta X\\ \Delta Y\\ \Delta Z\end{bmatrix} + \begin{bmatrix}1 & -\Delta k & \Delta\varphi\\ \Delta k & 1 & -\Delta\omega\\ -\Delta\varphi & \Delta\omega & 1\end{bmatrix}\begin{bmatrix}-\rho\sin(S\beta)\\ 0\\ -\rho\cos(S\beta)\end{bmatrix}\\ &= \vec{X}_0 + \begin{bmatrix}\Delta X\\ \Delta Y\\ \Delta Z\end{bmatrix} + \begin{bmatrix}1 & -\Delta k & \Delta\varphi\\ \Delta k & 1 & -\Delta\omega\\ -\Delta\varphi & \Delta\omega & 1\end{bmatrix}\begin{bmatrix}x\\ 0\\ -H\end{bmatrix}\end{aligned} \tag{26}$$

$$\begin{aligned}\vec{X}_G &\approx \vec{X}_0 + \begin{bmatrix}\Delta X\\ \Delta Y\\ \Delta Z\end{bmatrix} + \begin{bmatrix}-1 & \Delta k & -\Delta\varphi\\ -\Delta k & -1 & \Delta\omega\\ -\Delta\varphi & \Delta\omega & 1\end{bmatrix}\begin{bmatrix}-\rho\sin(S\beta)\\ 0\\ -\rho\cos(S\beta)\end{bmatrix}\\ &= \vec{X}_0 + \begin{bmatrix}\Delta X\\ \Delta Y\\ \Delta Z\end{bmatrix} + \begin{bmatrix}-1 & \Delta k & -\Delta\varphi\\ -\Delta k & -1 & \Delta\omega\\ -\Delta\varphi & \Delta\omega & 1\end{bmatrix}\begin{bmatrix}x\\ 0\\ -H\end{bmatrix}\end{aligned} \tag{27}$$

where,

Equation (26) is valid for a system flying along the positive direction of the Y-axis (this flight line will be denoted as the forward strip).

Equation (27) is valid for a system flying along the negative direction of the Y-axis (this flight line will be denoted as the backward strip).

$\Delta X, \Delta Y$ and ΔZ are the components of the lever-arm offset vector $\vec{P}_G$

S is the scale factor for the mirror angle β.

H is the flight height above ground, and

x is the lateral distance between the LiDAR point in question and the projection of the flight trajectory onto the ground.

The LiDAR point coordinates $\vec{X}_G$, as shown above, are function of the system parameter $\vec{x}$ and measurement $\vec{l}$ and represent the true point coordinates $\vec{X}_{True}$. Therefore, equation (26) and (27) can also be shown as:

$$\vec{X}_G = \vec{X}_{True} = f(\vec{x}, \vec{l}) \tag{28}$$

where,

$\vec{x} = (\Delta X, \Delta Y, \Delta Z, \Delta\omega, \Delta\varphi\ \Delta k)$, and

$\vec{l} = (\vec{X}_0, yaw, pitch, roll, \beta, \rho, S)$

In the presence of biases in the system parameters, the LiDAR point system will become biases $\vec{X}_{Biased}$ and will be function of the system parameter and measurements as well as biases in the system parameters $\delta\vec{x}$, as shown in equation (29).The lever-arm offset biases are represented by $(\delta\Delta X, \delta\Delta Y, \delta\Delta Z)$ and the boresight angle biases are represented by $(\delta\Delta\omega, \delta\Delta\varphi, \delta\Delta k)$.

$$\vec{X}_{Biased} = f(\vec{x} + \delta\vec{x}, \vec{l}) \tag{29}$$

where, $\delta\vec{x} = (\delta\Delta X, \delta\Delta Y, \delta\Delta Z, \delta\Delta\omega, \delta\Delta\varphi, \delta\Delta k)$

To analyze the impact of the biases, equation (29) is linearized with respect to the system parameters using a Taylor series expansion, yielding the form of equation (30) and (31), after ignoring the second and higher order terms. The term $\frac{\partial f}{\partial \vec{x}}$ represents the partial derivative with respect to the system parameters, while the term $\frac{\partial f}{\partial \vec{x}}\delta\vec{x}$ represents the impact of the system biases onto the derived point cloud $\delta\vec{X}_G$.

$$\vec{X}_{Biased} \approx f(\vec{x}, \vec{l}) + \frac{\partial f}{\partial \vec{x}}\delta\vec{x} = \vec{X}_{True} + \begin{bmatrix} \delta X_G \\ \delta Y_G \\ \delta Z_G \end{bmatrix} \tag{30}$$

$$\vec{X}_{Biased} \approx \vec{X}_{True} + \begin{bmatrix} \delta X_G \\ \delta Y_G \\ \delta Z_G \end{bmatrix}_{\delta\Delta X, \delta\Delta Y, \delta\Delta Z} + \begin{bmatrix} \delta X_G \\ \delta Y_G \\ \delta Z_G \end{bmatrix}_{\delta\Delta\omega, \delta\Delta\varphi, \delta\Delta k} = \vec{X}_{True} + \begin{bmatrix} \delta X_G \\ \delta Y_G \\ \delta Z_G \end{bmatrix}_{Total} \tag{31}$$

By differentiating the equation (26) and (27) with respect to the system parameters, the terms $\begin{bmatrix}\delta X_G\\ \delta Y_G\\ \delta Z_G\end{bmatrix}_{\delta\Delta X,\delta\Delta Y,\delta\Delta Z}$ and $\begin{bmatrix}\delta X_G\\ \delta Y_G\\ \delta Z_G\end{bmatrix}_{\delta\Delta\omega,\delta\Delta\varphi,\delta\Delta k}$ can be represented in the following equations:

$$\begin{bmatrix}\delta X_G\\ \delta Y_G\\ \delta Z_G\end{bmatrix}_{\delta\Delta X,\delta\Delta Y,\delta\Delta Z} = \frac{\partial f}{\partial \Delta X}\delta\Delta X + \frac{\partial f}{\partial \Delta Y}\delta\Delta Y + \frac{\partial f}{\partial \Delta Z}\delta\Delta Z = \begin{bmatrix}\pm\delta\Delta X\\ \pm\delta\Delta Y\\ \delta\Delta Z\end{bmatrix} \tag{32}$$

$$\begin{bmatrix}\delta X_G\\ \delta Y_G\\ \delta Z_G\end{bmatrix}_{\delta\Delta\omega,\delta\Delta\varphi,\delta\Delta k} = \frac{\partial f}{\partial \Delta\omega}\delta\Delta\omega + \frac{\partial f}{\partial \Delta\varphi}\delta\Delta\varphi + \frac{\partial f}{\partial \Delta k}\delta\Delta k = \begin{bmatrix}\mp H\delta\Delta\varphi\\ \pm H\delta\Delta\omega \pm x\delta\Delta k\\ -x\delta\Delta\varphi\end{bmatrix} \tag{33}$$

And therefore, the equation (31) would lead to the following equation:

$$\begin{bmatrix}\delta X_G\\ \delta Y_G\\ \delta Z_G\end{bmatrix}_{Total} = \vec{X}_{Biased} - \vec{X}_{True} = \begin{bmatrix}X_{Biased} - X_{True}\\ Y_{Biased} - Y_{True}\\ Z_{Biased} - Z_{True}\end{bmatrix} \approx \begin{bmatrix}\pm\delta\Delta X \mp H\delta\Delta\varphi\\ \pm\delta\Delta Y \pm H\delta\Delta\omega \pm x\delta\Delta k\\ \delta\Delta Z - x\delta\Delta\varphi\end{bmatrix} \tag{34}$$

In equation (32), (33) and (34), the multiple signs indicate the impact for the forward (equation (26)) and backward strips (equation (27)), with top sign referring to the forward strip. By subtracting the equation (34) from each other, the mathematical relationship between conjugate points in overlapping strips, which are flown in opposite or the same directions, can be derived.

3.2.1 Strips flown in Opposite Direction

The relationship for two strips (noted as strip A and strip B), which are flown in opposite directions is shown in equation (35) (36) and (37), where A is the forward strip and B is the backward strip. The configuration is shown in Figure 3-3

$$\begin{bmatrix}X_A - X_{True}\\ Y_A - Y_{True}\\ Z_A - Z_{True}\end{bmatrix} \approx \begin{bmatrix}\delta\Delta X - H\delta\Delta\varphi\\ \delta\Delta Y + H\delta\Delta\omega + x_A\delta\Delta k\\ \delta\Delta Z - x_A\delta\Delta\varphi\end{bmatrix} \tag{35}$$

$$\begin{bmatrix}X_B - X_{True}\\ Y_B - Y_{True}\\ Z_B - Z_{True}\end{bmatrix} \approx \begin{bmatrix}-\delta\Delta X + H\delta\Delta\varphi\\ -\delta\Delta Y - H\delta\Delta\omega - x_B\delta\Delta k\\ \delta\Delta Z - x_B\delta\Delta\varphi\end{bmatrix} \tag{36}$$

$$\begin{bmatrix} X_A - X_B \\ Y_A - Y_B \\ Z_A - Z_B \end{bmatrix} \approx \begin{bmatrix} 2\delta\Delta X - 2H\delta\Delta\varphi \\ 2\delta\Delta Y + 2H\delta\Delta\omega + (x_A + x_B)\delta\Delta k \\ -(x_A - x_B)\,\delta\Delta\varphi \end{bmatrix} \quad (37)$$

Since the goal of this research is to eliminate the horizontal strip discrepancies, only the first two rows in equation (37) are considered:

$$\begin{bmatrix} X_A - X_B \\ Y_A - Y_B \end{bmatrix} \approx \begin{bmatrix} 2\delta\Delta X - 2H\delta\Delta\varphi \\ 2\delta\Delta Y + 2H\delta\Delta\omega + (x_A + x_B)\delta\Delta k \end{bmatrix} \quad (38)$$

Equation (38) can be further simplified. The Figure 3-4 illustrates two cases of the backward strip B, where B_L is the backward strip on the left side of the forward strip A and B_R is the backward strip on the right side. Let D be the lateral distance between the two flight lines. In the case of B_L, it can be shown that $D = -(x_A + x_B)$ because both x_A and x_B are negative values. In the case of B_R, $D = (x_A + x_B)$. Therefore, the equation (38) can be written as equation (39) after simplification:

$$\begin{bmatrix} X_A - X_B \\ Y_A - Y_B \end{bmatrix} \approx \begin{bmatrix} 2\delta\Delta X - 2H\delta\Delta\varphi \\ 2\delta\Delta Y + 2H\delta\Delta\omega \mp D\delta\Delta k \end{bmatrix} \quad (39)$$

where multiple signs represent the two cases B_L and B_R, with the top sign referring to B_L

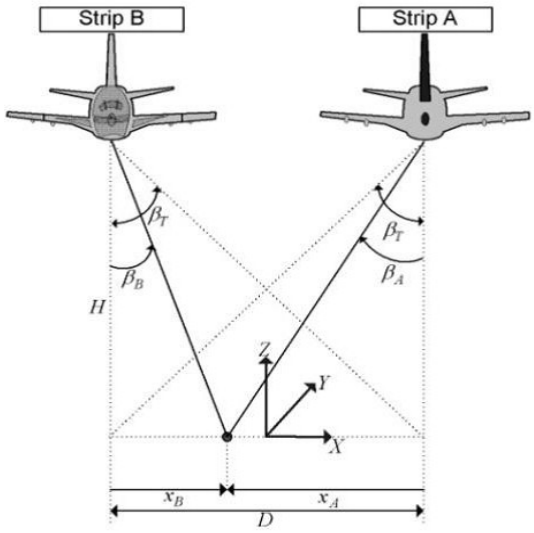


Figure 3-3 Conjugate Point in Overlapping Strips Flown in Opposite Directions [2]

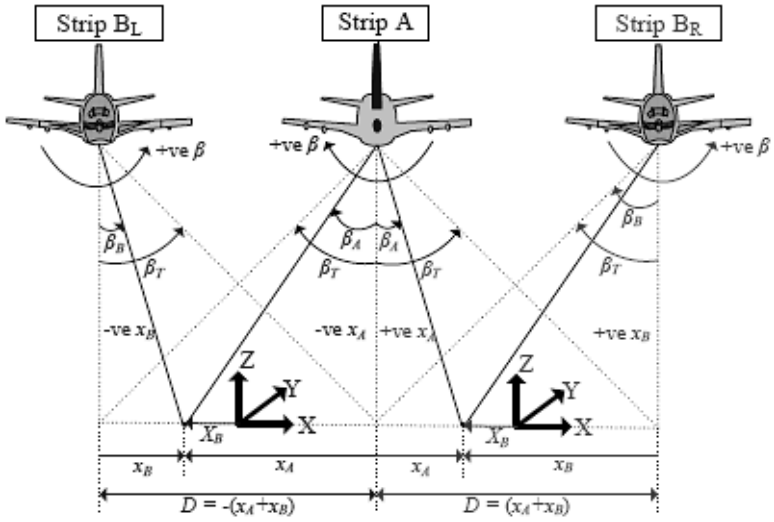


Figure 3-4 Two Cases of the Backward Strip [2]

3.2.2 Strips flown in the Same Direction

Following a similar procedure, the relationship between two overlapping strips (noted as strip A and strip B again), which are flown in the same directions, is shown in equation (40) with all the intermediate steps skipped. The configuration is shown in Figure 3-5.

$$\begin{bmatrix} X_A - X_B \\ Y_A - Y_B \end{bmatrix} \approx \begin{bmatrix} 0 \\ -D\delta\Delta k \end{bmatrix} \tag{40}$$

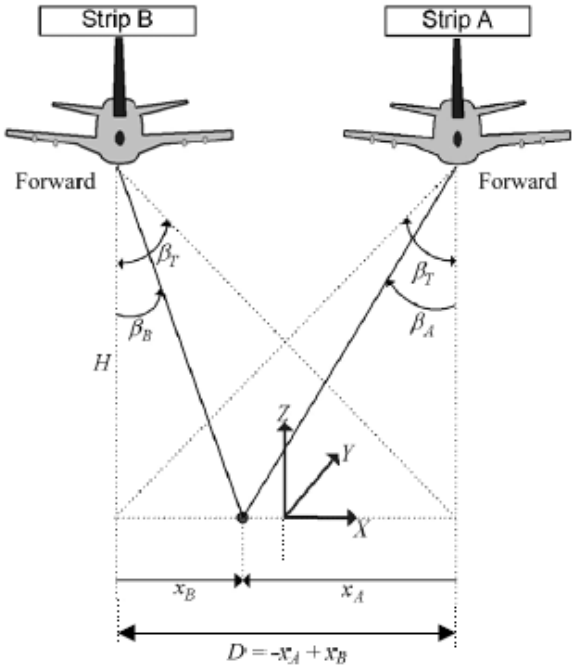


Figure 3-5 Conjugate Point in Overlapping Strips Flown in the Same Directions [2]

3.3 Proposed LiDAR Strip Mosaicing Algorithm

In this section, A LiDAR strip mosaicing algorithm is proposed. The algorithm is based on the mathematical model of the systematic biases in the previous section; therefore, it must be utilized under the corresponding conditions for the model as well. For simplicity and clearance, the algorithm is explained though the process of mosaicing three parallel overlapping LiDAR strips, noted as A, B and C. As shown in Figure 3-6, the strip A overlaps the strip B and the strip B overlaps the strip C. The l_A, l_B and l_C are the flight lines in the middle of each strip.

Two scenarios are considered in the algorithm. In the first scenario, the strip A and Strip C are flown in the backward direction and the strip B is flown in the forward direction. This scenario is noted as Backward-Forward-Backward (BFB) strip configuration and represents the practical case when multiples strips are flown in

opposite directions alternatively. In the second scenario, all the strips are flown in the forward direction. This scenario is noted as Forward-Forward-Forward (FFF) strip configuration and represents the case when all the strips are flown in the same directions.

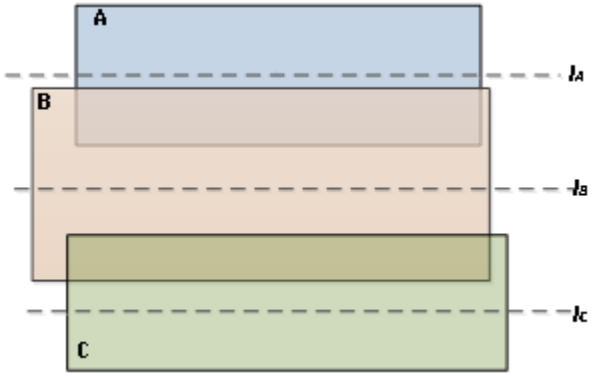


Figure 3-6 Configuration of Three LiDAR Strips

3.3.1 BFB Strip Configuration

Assuming that the direction of $+Y$ axis is from left to right in Figure 3-6, the strip A is equivalent to the B_R strip and strip C is equivalent to the B_L strip. Using the equation (39), biases between strip pairs of BA and BC are shown as:

$$\begin{bmatrix} X_B - X_A \\ Y_B - Y_A \end{bmatrix} \approx \begin{bmatrix} 2\delta\Delta X - 2H\delta\Delta\varphi \\ 2\delta\Delta Y + 2H\delta\Delta\omega - D_{BA}\delta\Delta k \end{bmatrix} \tag{41}$$

$$\begin{bmatrix} X_B - X_C \\ Y_B - Y_C \end{bmatrix} \approx \begin{bmatrix} 2\delta\Delta X - 2H\delta\Delta\varphi \\ 2\delta\Delta Y + 2H\delta\Delta\omega + D_{BC}\delta\Delta k \end{bmatrix} \tag{42}$$

where, D_{BA} is the distance between l_A and l_B. D_{BC} is the distance between l_C and l_B

For simplicity , the following notation are used:

$$\begin{bmatrix} X_T^{AB} \\ Y_T^{AB} \end{bmatrix} = \begin{bmatrix} X_B - X_A \\ Y_B - Y_A \end{bmatrix} \tag{43}$$

$$dX = \delta\Delta X - H\delta\Delta\varphi \quad \text{and} \quad dY = \delta\Delta Y + H\delta\Delta\omega \tag{44}$$

Then the equation (41) and (42) will lead to equation (45) and (46)

$$\begin{bmatrix} X_T^{AB} \\ Y_T^{AB} \end{bmatrix} = \begin{bmatrix} 2dX \\ 2dY - D_{BA}\delta\Delta k \end{bmatrix} \tag{45}$$

$$\begin{bmatrix} X_T^{BC} \\ Y_T^{BC} \end{bmatrix} = \begin{bmatrix} 2dX \\ 2dY + D_{BC}\delta\Delta k \end{bmatrix} \tag{46}$$

Subtracting the second row of equation (45) from (46) would lead to equation

$$Y_T^{AB} - Y_T^{BC} = (D_{BC} + D_{BA})\delta\Delta k \tag{47}$$

Therefore,

$$\delta\Delta\hat{k} = \frac{Y_T^{AB} - Y_T^{BC}}{D_{BC} + D_{BA}} \tag{48}$$

where $\delta\Delta\hat{k}$ is the estimation of the system boresight roll angle bias $\delta\Delta k$

As shown in the equation above, the estimated bias in the boresight yaw angle $\delta\Delta\hat{k}$ can be directly calculated in this strip configuration. The variables Y_T^{AB} and Y_T^{BC} represent the strip discrepancies between AB and BC along the Y-axis. After the $\delta\Delta\hat{k}$ is calculated, the estimated value $d\hat{X}$ and $d\hat{Y}$ can be calculated as

$$d\hat{X} = \frac{X_T^{AB} + X_T^{BC}}{4} \tag{49}$$

$$d\hat{Y} = \frac{(Y_T^{AB} + Y_T^{BC}) - (D_{BC} - D_{BA})}{4\delta\Delta\hat{k}} \tag{50}$$

Recall the equations (35) and (30), which describes the discrepancies between the true coordinates and biased coordinates. By introducing the definition of dX and dY from equation (44), the estimated true values can be then calculated by equation (51) for forward strip and (52) for backward strip.

$$\begin{bmatrix} \hat{X}_{True} \\ \hat{Y}_{True} \end{bmatrix} = \begin{bmatrix} X_{Biased} - d\hat{X} \\ Y_{Biased} - d\hat{Y} - \hat{x}\delta\Delta\hat{k} \end{bmatrix} \tag{51}$$

$$\begin{bmatrix} \hat{X}_{True} \\ \hat{Y}_{True} \end{bmatrix} = \begin{bmatrix} X_{Biased} + d\hat{X} \\ Y_{Biased} + d\hat{Y} + \hat{x}\delta\Delta\hat{k} \end{bmatrix} \tag{52}$$

where the $\hat{x}$ is determined approximately by the distance from the biased point to the flight line.

Equations (51) and (52) indicate that a complete horizontal LiDAR strip adjustment can be achieved in the BFB strip configuration. Although only one of the system bias parameters ($\delta\vec{x}$) is directly calculated, the true coordinates can still be estimated as long as $\delta\Delta\hat{k}, d\hat{X}$ and $d\hat{Y}$ are available.

In order to calculate $\delta\Delta\hat{k}, d\hat{X}$ and $d\hat{Y}$, the values of X_T^{AB}, X_T^{BC}, Y_T^{AB} and Y_T^{BC}, which are the strip discrepancies along X-axis and Y-axis for each pair of strips, must be obtained. By applying the ARFD-based LiDAR strip matching method, the homography matrix between each strip pair can be calculated. Let H_{BA} be the corresponding 3×3 homography matrix from the strip A to strip B. If an affine transform is assumed, strip A and strip B can be related as:

$$\begin{bmatrix} X_B \\ Y_B \\ 1 \end{bmatrix} = H_{BA} \begin{bmatrix} X_A \\ Y_A \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} X_A \\ Y_A \\ 1 \end{bmatrix} \tag{53}$$

where,

$\boldsymbol{R}$ is the 2×2 rotational matrix.

$\boldsymbol{T}$ is the 2×1 translation vector.

Equation (40) shows that for every conjugate point pair in the overlapping area, the horizontal bias is $\begin{bmatrix} 2\delta\Delta X - 2H\delta\Delta\varphi \\ 2\delta\Delta Y + 2H\delta\Delta\omega \mp D\delta\Delta k \end{bmatrix}$. Since all the parameters involved are system constants, the transformation between strip A and strip B can be represented by pure translation. Therefore, the H_{BA} can be expressed as:

$$H_{BA} = \begin{bmatrix} 1 & 0 & T_X \\ 0 & 1 & T_Y \\ 0 & 0 & 1 \end{bmatrix} \tag{54}$$

$$T_X = X_B - X_A = X_T^{AB} \tag{55}$$

$$T_Y = Y_B - Y_A = Y_T^{AB} \tag{56}$$

Once the Homography matrix H_{BA} is obtained by applying the ARFD-based LiDAR strip matching algorithm, the horizontal strip discrepancies X_T^{AB} and Y_T^{AB} can be found by extracting the translation vector from the H_{BA}. Following the same procedure, the X_T^{BC} and Y_T^{BC} can be found from H_{BC}. After finding these four parameters, $\delta\Delta\hat{k}, d\hat{X}$ and $d\hat{Y}$ can be calculated using equation (48), (49) and (50). Finally, for every point in the strips, its true coordinates can be estimated by equation (51) and (52).

Since the true coordinates with respect to the ground coordinate system are estimated in this scenario, the global alignment errors are eliminated automatically and the common ground, with respect to which all points are registered, is the ground coordinate system itself. By interpolating all the estimated LiDAR points together, the LiDAR strip mosaic task is implicitly accomplished.

3.3.2 FFF Strip Configuration

Similar to BFB strip configuration, the direction of $+Y$ axis is set from left to right, again. In this case, equation (40) is used to represent the biases between the strip pair AB and BC:

$$\begin{bmatrix} X_B - X_A \\ Y_B - Y_A \end{bmatrix} \approx \begin{bmatrix} 0 \\ -D_{BA}\delta\Delta k \end{bmatrix} \tag{57}$$

$$\begin{bmatrix} X_C - X_B \\ Y_C - Y_B \end{bmatrix} \approx \begin{bmatrix} 0 \\ -D_{BC}\delta\Delta k \end{bmatrix} \tag{58}$$

Again, the notation in equation (43) is used for simplicity. Then the equation (57) and (58) will lead to equation (59) and (60)

$$\begin{bmatrix} X_T^{AB} \\ Y_T^{AB} \end{bmatrix} = \begin{bmatrix} 0 \\ -D_{BA}\delta\Delta k \end{bmatrix} \tag{59}$$

$$\begin{bmatrix} X_T^{CB} \\ Y_T^{CB} \end{bmatrix} = \begin{bmatrix} 0 \\ -D_{BC}\delta\Delta k \end{bmatrix} \tag{60}$$

Adding up the second row of equation (59) and (60) helps $\delta\Delta\hat{k}$ as

$$\delta\Delta\hat{k} = -\frac{Y_T^{AB} + Y_T^{CB}}{D_{BA} + D_{BC}} \tag{61}$$

It is obvious that the estimation of $\delta\Delta\hat{k}$ actually does not need more than two strips. For example, without the presence of strip C, the $\delta\Delta\hat{k}$ can be calculated by $\delta\Delta\hat{k} = \frac{Y_T^{AB}}{D_{BA}}$. The reasons of using 3 strips are:

1. To be consistent with the BFB strip configuration where 3 strips are used.
2. Allowing equation (61) which gives an average value of $\delta\Delta\hat{k}$ by using all strips, thereby reducing the global error.

Similar to the steps in the BFB scenario, the value of X_T^{AB}, X_T^{BC}, Y_T^{AB} and Y_T^{BC} can be found from the homography matrices H_{BA} and H_{CB} in order to calculate $\delta\Delta\hat{k}$. However, the estimates of dX and dY from BFB are not applicable to FFF, as can be seen by comparing the equation pairs (45), (46) for BFB and (59), (60) for FFF. Thus estimation of the true coordinates is not possible for FFF. Despite this, the strip mosaic can still be achieved indirectly.

Consider the first two rows of equation (35), which describes the relationship between the true coordinate and biased coordinate. If the parameter positions are rearranged on both sides of the equality sign, the equation (35) can be re-written as:

$$\begin{bmatrix} X_{True} + (\delta\Delta X - H\delta\Delta\varphi) \\ Y_{True} + (\delta\Delta Y + H\delta\Delta\omega) \end{bmatrix} = \begin{bmatrix} X_{Biased} \\ Y_{Biased} - x\delta\Delta k \end{bmatrix} \tag{62}$$

Using the notation in equation (44) leads to

$$\begin{bmatrix} X_{True} + dX \\ Y_{True} + dY \end{bmatrix} = \begin{bmatrix} X_{Biased} \\ Y_{Biased} - x\delta\Delta k \end{bmatrix} \tag{63}$$

Introducing two new terms X'_{True} and Y'_{True} which are defined as

$$X'_{True} = X_{True} + dX \quad \text{and} \quad Y'_{True} = Y_{True} + dY \tag{64}$$

the estimated value of X'_{true} and Y'_{true} can be calculated by the following equation:

$$\begin{bmatrix} \hat{X}'_{True} \\ \hat{Y}'_{True} \end{bmatrix} = \begin{bmatrix} X_{Biased} \\ Y_{Biased} - \hat{x}\delta\Delta\hat{k} \end{bmatrix} \quad (65)$$

where the $\hat{x}$ is determined approximately by the distance from the biased point to the flight line and $\delta\Delta\hat{k}$ can be calculated though equation (61).

Since $\begin{bmatrix} X_{True} \\ Y_{True} \end{bmatrix}$ is the point registered with respect to the ground coordinate system, $\begin{bmatrix} \hat{X}'_{True} \\ \hat{Y}'_{True} \end{bmatrix}$ can be viewed as the point registered with respect to a shifted version of ground coordinate system with the shift equal to $\begin{bmatrix} dX \\ dY \end{bmatrix}$. It is important to note that for every estimated point $\begin{bmatrix} \hat{X}'_{True} \\ \hat{Y}'_{True} \end{bmatrix}$, the shift $\begin{bmatrix} dX \\ dY \end{bmatrix}$ is a constant. Therefore, by interpolating all the estimated points together, the global alignment errors can be eliminated but the common reference becomes a shifted version of the ground coordinate system. Since the shift is constant along both X and Y axis, its effect on the LiDAR strip mosaic can be modeled as a pure translation, with all the strips translated by the same unit. This translation will not be noticeable in the image domain.

Chapter 4

Experiment and Implementation and Results

In this chapter, we describe the details of the experiments that were used to validate our thesis algorithm related to LiDAR strip mosaicing. The implementation requires several steps. Before implementing the algorithm, the initial data set needs to be preprocessed. Next, two LiDAR strip mosaicing methods are implemented with different bias parameters and the corresponding results are discussed. The overall experiment is implemented on MATLAB 2013a with the aid of the Image Processing Toolbox from MathWorks Inc.

4.1 LiDAR Data Preprocessing

The goal of the data preprocessing is to output the pair-wise homography matrices for further strip mosaicing. Firstly, simulated systematic biases are added to the initial LiDAR dataset and then the modified ARFD-based LiDAR image-matching algorithm is applied on every pair of overlapping strips in order to obtain the homography matrices for each pair.

4.1.1 Initial LiDAR Dataset

The dataset used for the experiment is obtained from the National Resource Conversation Service (NRCS), which is a part of the US Department of Agriculture. The details of the LiDAR data are the following:

1. Covered area: Saginaw Bay, Michigan, USA.
2. Flight height: 6500 feet
3. Footprint spacing: 0.7m.
4. Vertical accuracy: 0.135m at a 95% confidence level
5. Horizontal accuracy: ± 3.8 foot (1.16m) at a 95% confidence level.

6. Flight Directional Information: The swaths are flown in the east-west directions, therefore the Y-axis is along east-west and X-axis is along north-south. The direction of $+Y$ is manually selected to be toward west and the direction of $+X$ is toward the north.
7. Number of swaths: A set of 9 flight swaths is available in LiDAR points cloud data. Each swath contains millions of points and each point contains four values X, Y, Z and I, which are the 3D coordinates and the intensity measurement.

Since each swath covers a large geographic area, small sub-patches of rectangular shape, also called verticals, were extracted, as shown in Figure 4-1. Because some swaths are smaller than the others thereby offering lesser number of strips, not all verticals contain all the 9 strips. In this experiment, vertical 1, which contains 5 strips, and vertical 2, which contains 7 strips, are utilized.

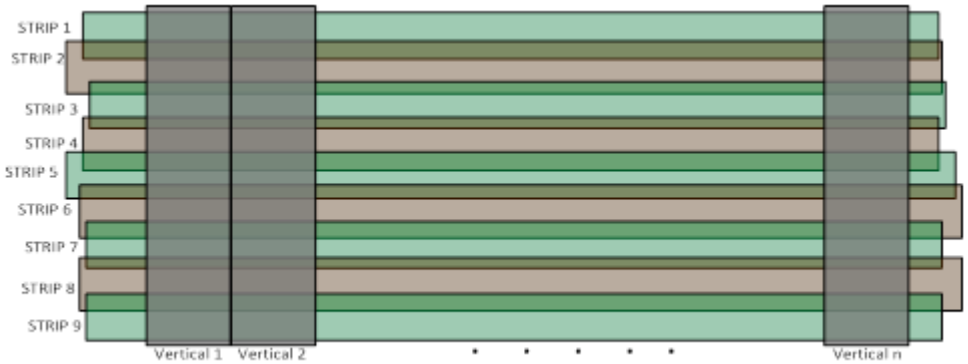


Figure 4-1 Patch Extraction from Individual Strips [9]

4.1.2 Adding Simulated System Biases

The data was calibrated by a vendor company using a proprietary method. At the virtual environment lab in UTA, interpolated images of the vertical 1 and vertical 2 are firstly generated in order to visualize the original data, as shown in Figure 4-2 and Figure

4-3, because LiDAR points are irregularly sampled and stored as point cloud. The intensity data is used during interpolation due to its high contrast. As the footprint spacing is $0.7m$, the 1D resolution for interpolation is selected to be $0.7m/pixel$, meaning each pixel occupies an area of size equal to $0.7m \times 0.7m = 0.49m^2$. Besides, in the Figure 4-2, an area on the boundary of two strips is enlarged to show details, where a road across two strips is considered as an indicator of strip discrepancies. Since there are not any noticeable horizontal discrepancies between the strips and the road in the enlarged area is straight, the initial dataset is regarded as the true data $\vec{X}_{True}$.

To conduct the experiment, systematic biases are added to the initial dataset by using equation (34) to generate the simulated biases that might have contaminated data cloud before the vendor processed it. The initial system biases parameters $(\delta\Delta X, \delta\Delta Y, \delta\Delta Z, \delta\Delta\varphi, \delta\Delta k, \delta\Delta\omega)$ are arbitrarily selected. The flight height $H = 6500 \text{ feet} = 1981.2\text{m}$ is given in the metadata. In order to obtain x, which is the lateral distance between the LiDAR point in question and the projection of the flight line onto the ground, the flight line should be determined initially. Let $\vec{X} = (X, Y, Z)$ denote 3D coordinate of a LiDAR point with respect to the ground coordinate system. Since the flight line is in the middle of each swath, the position of the middle line X_{middle} can be calculated by taking the average value of X-axis coordinates of the topmost point $\vec{X}_{top}$ and the bottommost point $\vec{X}_{bottom}$, which can be mathematically expressed as $X_{middle} = \frac{X_{top} + X_{bottom}}{2}$. Then the x value of each point $\vec{X}$ is equivalent to $(X - X_{middle})$ for forward strip and $-(X - X_{middle})$ for backward strip. After the value of x is determined, bias can be added to the original data by following equation (34).

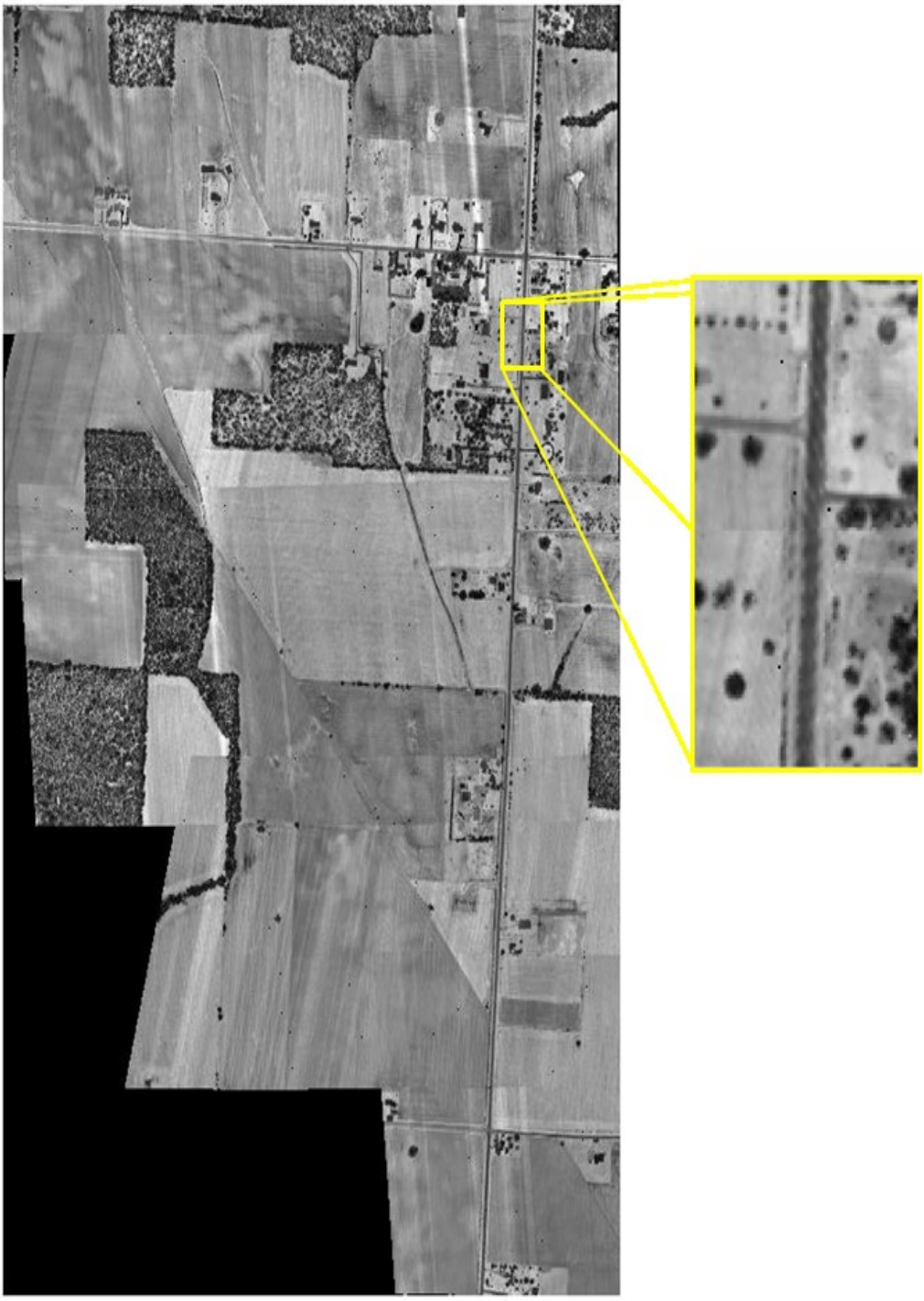


Figure 4-2 Original Vertical 1 with an Enlarged Area

Figure 4-3 Original Vertical 2

4.1.3 Modified ARFD based Image Matching Algorithm

A modified ARFD based Image matching algorithm is applied on the data cloud (with simulated biases added) in order to get the homography matrices. Apart from the original algorithm introduced in the Chapter 2, the modifications are described in the following.

4.1.3.1 Interpolation with User-defined Fixed resolution

The interpolation process is modified to output LiDAR images with user-defined fixed resolution instead of the non-fixed resolution in the original implementation. In the interpolation step, the range of the strip is firstly determined by finding the leftmost, rightmost, topmost and bottommost point within the strip. The four points are noted as $\vec{X}_{left}, \vec{X}_{right}, \vec{X}_{top}$ and $\vec{X}_{bottom}$. The 1D resolution (res) is manually selected. The number of pixels along each row (N_r) is equivalent to $\left\lceil \frac{Y_{left} - Y_{right}}{res} \right\rceil$ and the number of pixels along each column (N_c) is equivalent to $\left\lceil \frac{X_{top} - X_{bottom}}{res} \right\rceil$, where $\lceil\ \rceil$ denotes the ceiling operation. Then the raster of size $N_c \times N_r$ is create by starting at the position (X_{bottom}, Y_{left}) with the increment equal to res along both row and column directions.

4.1.3.2 Using Intensity Data for Keypoints Extraction

The original algorithm has shown that using intensity data to generate key points leads to very low matching performance [9]. However, in some cases, very few key points can be extracted from the elevation image if the strip is flown over large flat region, like crop fields. Since at least 4 correspondences are required to calculate the homography matrix, the intensity data may have to be used to extract more key points. Figure 4-4 shows the comparison of the extracted keypoint pairs by different method between two adjacent strips. Each keypoint pair is connected by a straight line from one strip to the

other. It is obvious that using both elevation and intensity data (Figure 4-4 (b)) gives more keypoint pairs than using elevation data only (Figure 4-4 (a))

4.1.3.3 Pruning and Re-selecting Inliers by Using Spatial Information

Since key points from intensity data are used, several outliers could result among the matched points. Depending on the number of outliers, evaluating the degenerate cases in RANSAC can lead to erroneous homography matrix [40]. In order to obtain a precise homography matrix, an additional inlier pruning and re-selecting algorithm is added after the RANSAC. The basis of the additional pruning algorithm is described below.

The system bias model implies that the transformation between the overlapping strips is purely translational. This information could be used to reduce the outliers and increase the inliers. The inlier pruning and re-selecting algorithm could be generalized in the following steps:

1. After the RANSAC algorithm, the number of correspondences extracted from elevation data in the inliers is counted. If the number is greater than an arbitrary threshold (set as 2 in the implementation), go to step 2; otherwise go to step 3.
2. Calculate the translation vectors of the inliers, which are extracted only from the elevation data. Determine both the magnitude and the direction of each translation vector. Determine the average and standard deviation of the magnitudes and the average of directions.
3. Calculate the translation vectors of the inliers, which are extracted from both the elevation and intensity data. Determine both the magnitude and the direction of each translation vector. Determine the average and standard deviation of the magnitudes and the average of directions.

4. The three values either from step 2 or step 3 are noted as $L_m, \sigma \; and \; \theta_m$. Calculate the translation vector of every correspondence in both inliers and outliers. Calculate the magnitude and direction of each translation vector as well. For each translation vector V_i, its magnitude is L_i and direction is θ_i
5. Evaluate each pair of (L_i, θ_i). If both conditions $(L_m - \sigma) \leq L_i \leq (L_m + \sigma)$ and $|\theta_i - \theta_m| \leq 1.5^o$ are satisfied, the corresponding point correspondence is picked as inlier; otherwise, it becomes an outlier.
6. Repeat step 5 for every vector V_i

Finally, the algorithm will pick the correspondences, which follow the translational model and reject most of the outliers. Figure 4-4 (c) shows the keypoint pairs extracted by using both elevation and intensity data with pruning and re-selection technique. It shows that much more keypoints are extracted. Figure 4-5 shows the comparison of strip matching result by using keypoint pairs from Figure 4-4 (b) and Figure 4-4 (c), respectively. It shows that the technique of pruning and re-selecting keypoints by using spatial information can help improve the matching result.

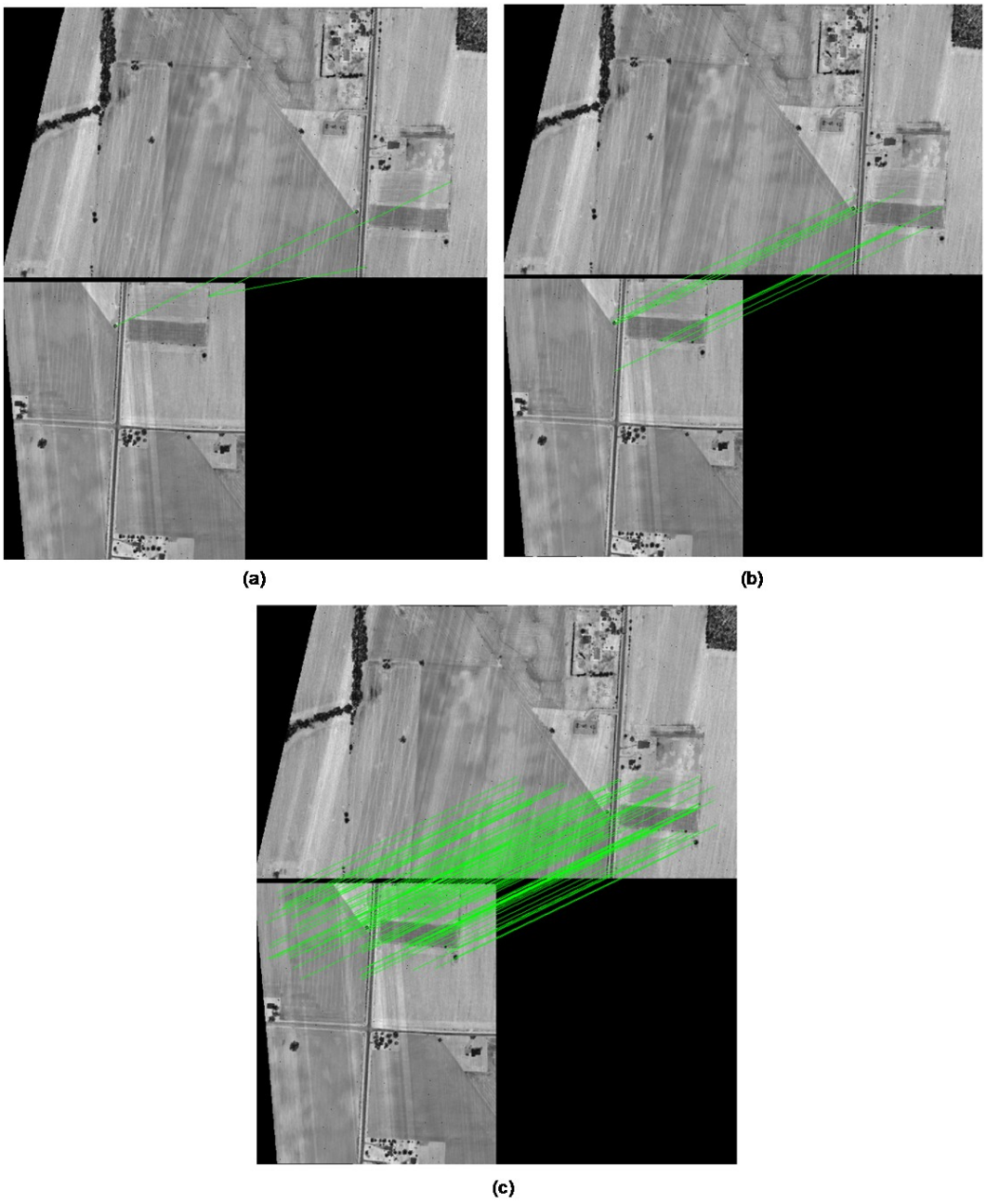
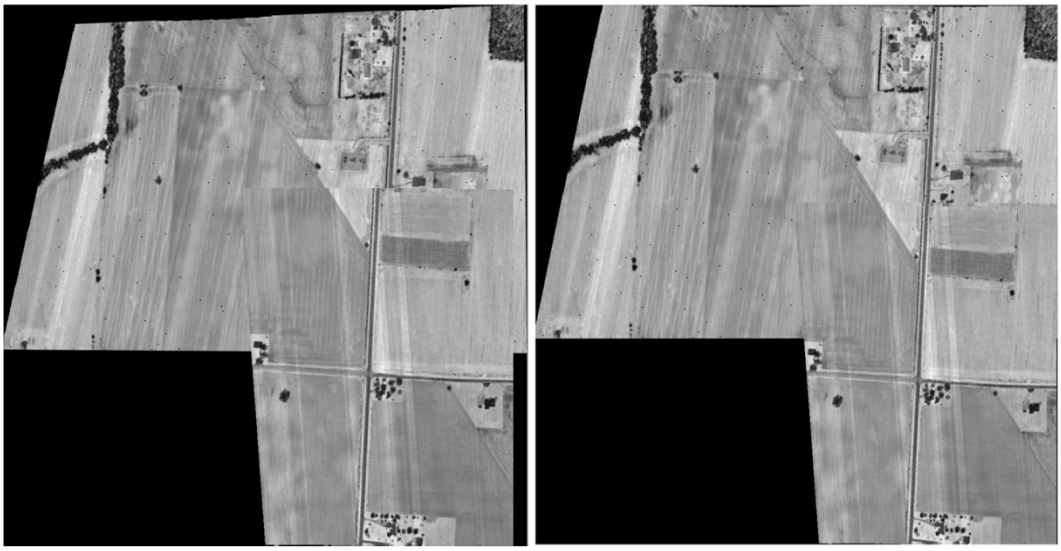


Figure 4-4 Comparison of Keypoints Extracted by (a) Using Elevation Data Only (b) Using Both Elevation and Intensity Data (c) Using Both Elevation and Intensity Data with Pruning and Re-selection



(a) (b)

Figure 4-5 Comparison of Strip Matching Result by (a) Using Keypoint Pairs from *Figure 4-4 (b)* and (b) Using Keypoints Pairs from *Figure 4-4 (c)*

4.2 Pair-wise LiDAR strip Mosaicing

In this section, the pair-wise LiDAR strip mosaicing method and result are presented. This method is based on Section 3.1.2.1, where the homography matrix between the common reference and each strip is calculated by chaining the pair-wise homography matrices.

4.2.1 Preprocessing Parameters

Vertical 1 (Figure 4-2) is used in this experiment. The five strips inside Vertical 1 are numbered 1,2,3,4 and 5 from top to bottom. The value of the system biases parameters are chosen to be very large in order to demonstrate biases clearly and the values are $\delta\Delta X = \delta\Delta Y = \delta\Delta Z = 2.5m,\ \delta\Delta\varphi = \delta\Delta k = \delta\Delta\omega = 1^o$. The strip configuration is selected to be the FFF pattern, where all the five strips are flown forward. The 1D resolution for interpolation is still 0.7m. The interpolation of the biased data is shown in Figure 4-6 with the same area enlarged as in Figure 4-2. By implementing the modified

ARFD-based LiDAR strip matching algorithm, four pair-wise homography matrix are obtained, which are H_{21}, H_{32}, H_{43} and H_{54}.

4.2.2 Results and Conclusion

It is obvious that in the Figure 4-6, the simulated biases cause the horizontal discrepancies between all the strips, as the straight road in the enlarged area of Figure 4-2 becomes discontinuous. In the pair-wise mosaicing method, the strip 5 is selected as the common reference. Therefore, the homography matrix relating each strip except strip 5 can be calculated by chaining homography matrices. For example, the homography matrix, relating strip 4 is H_{54}; the homography matrix relating strip 3 is equivalent to $H_{54}H_{43}$; and the homography matrix relating to strip 2 is $H_{54}H_{43}H_{32}$. The coordinate of every point within each strip is transformed by applying equation (19). Finally, all the transformed points are interpolated together in the mosaic space, with the common reference strip 5. The interpolated image mosaic is shown in Figure 4-7 with the area enlarged.

Figure 4-7 shows that the horizontal discrepancies between all the strips are eliminated, as the discontinuous road in Figure 4-6 becomes straight again. However, comparing the Figure 4-7 with Figure 4-2, it is obvious that the overall strip mosaic of Figure 4-7 is a sheared version of Figure 4-2. The shearing effect originates from the discrepancies between strip 5 and the ground coordinate system. Since strip 5 is chosen as the common ground, the shearing effect then propagates over all the strips after mosaicing. The result reveals the disadvantages of the pair-wise mosaicing method, implying that the pair-wise strip mosaicing method is not able to fully solve the LiDAR strip mosaicing problem

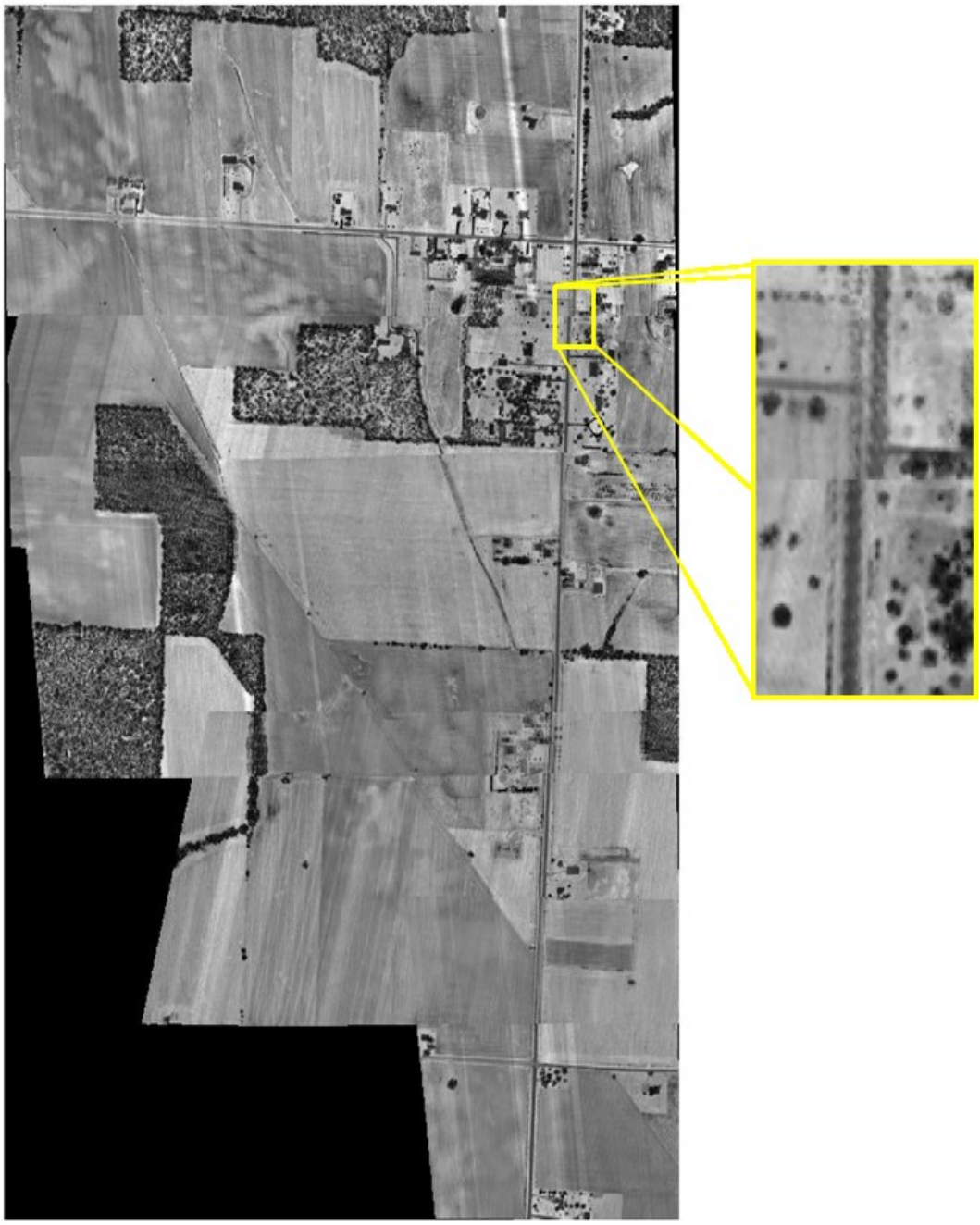


Figure 4-6 Simulated Biases-Added Vertical 1 with the Same Enlarged Area under FFF Strip Configuration

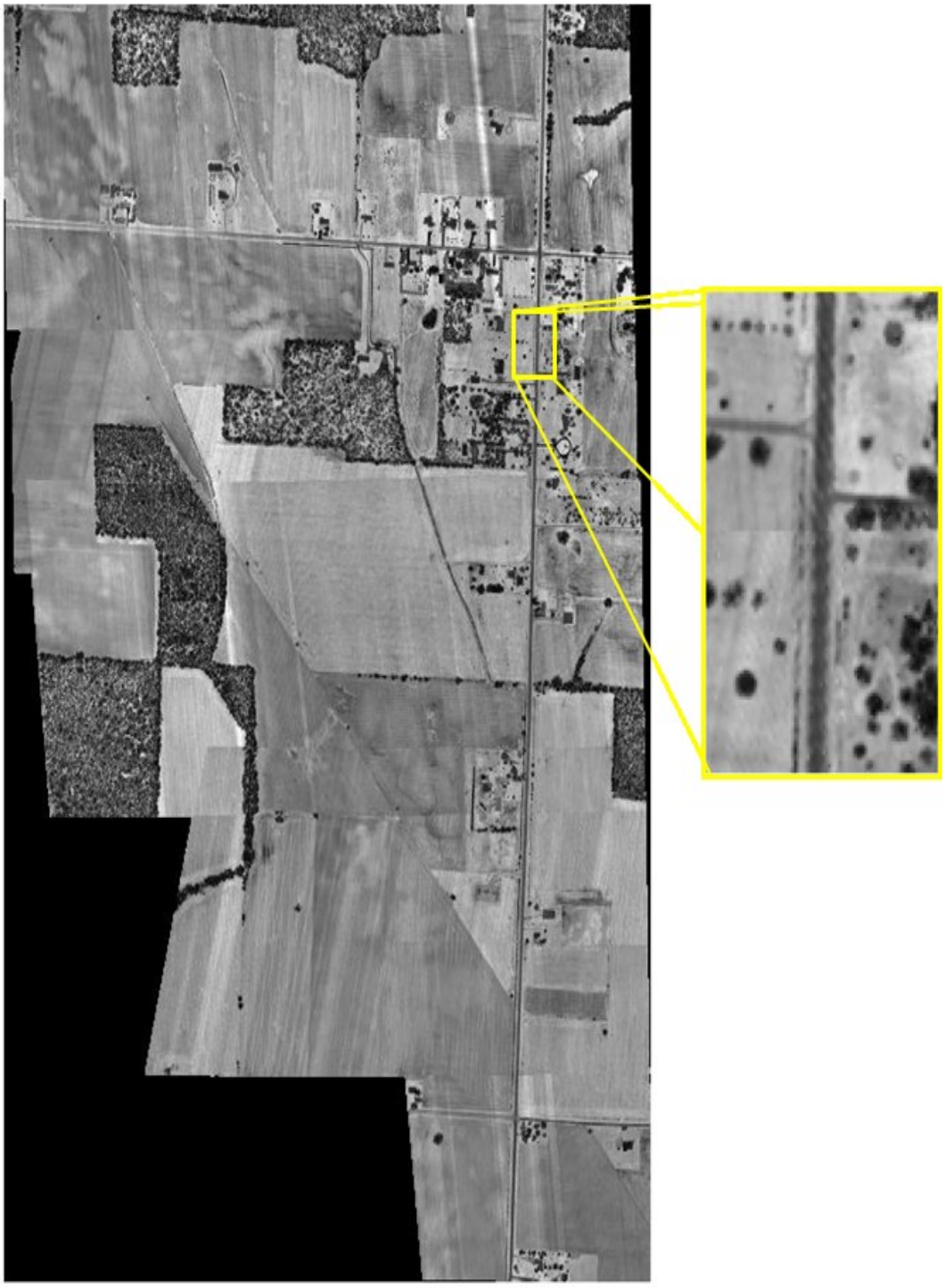


Figure 4-7 Pair-wise Strip Mosaicing Result with the Same Enlarged Area

4.3 Systematic-Bias-Model-Based LiDAR strip Mosaicing

In this section, the systematic-bias-model-based LiDAR strip mosaicing method and results are presented. In addition, the performance of this method is quantitatively evaluated under various values of systematic biases.

4.3.1 Preprocessing Parameters

Both Vertical 1 and Vertical 2 are used in this experiment and tested multiple times under various bias parameters. For simplicity, only Vertical 1 is used for the following explanation. Similarly, the five strips inside Vertical 1 are numbered 1,2,3,4 and 5 from top to bottom. Both BFB and FFF strip configuration are considered in this experiment. The system bias parameters are chosen from multiple sets of candidate values. Let the lever-arm offset biases $\delta\Delta X, \delta\Delta Y \text{ and } \delta\Delta Z$ always have equal value ($\delta\Delta X = \delta\Delta Y = \delta\Delta Z$) and so do the boresight angle biases $\Delta\varphi, \delta\Delta k \text{ and } \delta\Delta\varphi$ ($\delta\Delta\varphi = \delta\Delta k = \delta\Delta\omega$).

The set of candidate values for lever-arm offset biases is $\{0.05m, 0.6m, 1.2m, 2\text{m}\}$; the set of candidate values for boresight angle offset biases is $\{50'', 5', 30'\}$; and the set of candidate values for 1D interpolation resolution is $\{0.4m/pixel, 0.7m/pixel, 1.5m/pixel\}$. The choice of all the candidate values depends on the nominal pulse spacing, which is equal to 0.7m. For example, out of all the four candidate values of the lever-arm offset biases, $0.05m$ is much less than the nominal pulse spacing while $0.6m, 1.2m \text{ and } 2\text{m}$ are approximately equal to, twice as large as and three times as large as the nominal pulse spacing. Then, a single test is conducted by individually choosing one value from each of the 3 sets under either of the two strip configuration. Therefore, a total of $2 \times 4 \times 3 \times 3 = 72$ tests have been conducted. Figure 4-8 shows the block diagram of the workflow of each test. Detail of each block can be found in previous sections.

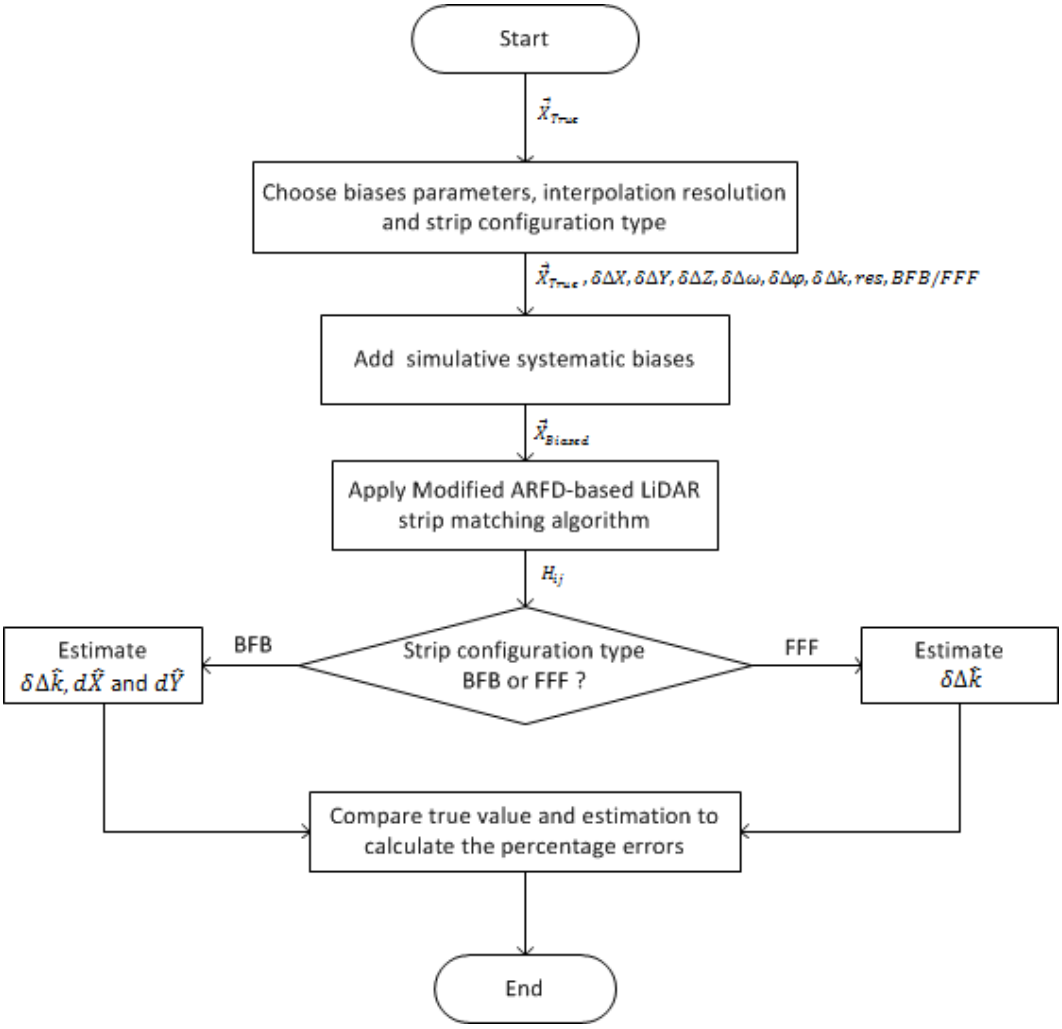


Figure 4-8 Block Diagram of the Workflow of Each Test

4.3.2 Results and Conclusion

By following the block diagram in Figure 4-8, the estimated values ($\{\delta\Delta\hat{k}, d\hat{X}, d\hat{Y}\}$ in the BFB strip configuration and $\delta\Delta\hat{k}$ in the FFF strip configuration) can be obtained and then used to estimate the true value of the coordinate of each point ($\begin{bmatrix}\hat{X}_{True}\\ \hat{Y}_{True}\end{bmatrix}$ in the BFB strip configuration and $\begin{bmatrix}\hat{X}'_{True}\\ \hat{Y}'_{True}\end{bmatrix}$ in the FF configuration). Finally, the reconstructed strip

mosaic image can be obtained by interpolating the estimated true value. Figure 4-9 shows the strip mosaic result with the same area enlarged, by using the systematic-bias-model-based LiDAR strip mosaicing algorithm. The strips are under the same simulated system biases as in Figure 4-6. As the result shows, there is not any noticeable difference between the mosaic result and the original data (Figure 4-2), indicating the proposed algorithm performs very well under the corresponding systematic bias parameters.

Figure 4-10 Shows the Vertical 1 after adding simulated biases under the BFB strip configuration. The bias parameters are $\delta\Delta X = \delta\Delta Y = \delta\Delta Z = 2.5m$, $\delta\Delta\varphi = \delta\Delta k = \delta\Delta\varphi = 1^o$ and the interpolation resolution is equal to $0.7m/pixel$. Figure 4-11 shows the final strip mosaic by using the proposed method. Comparing Figure 4-11 with the original data (Figure 4-2) shows again there is not any noticeable difference between the two mosaics.

Figure 4-9 Systematic-Bias-Model-Based Strip Mosaicing Result with the Same Enlarged Area

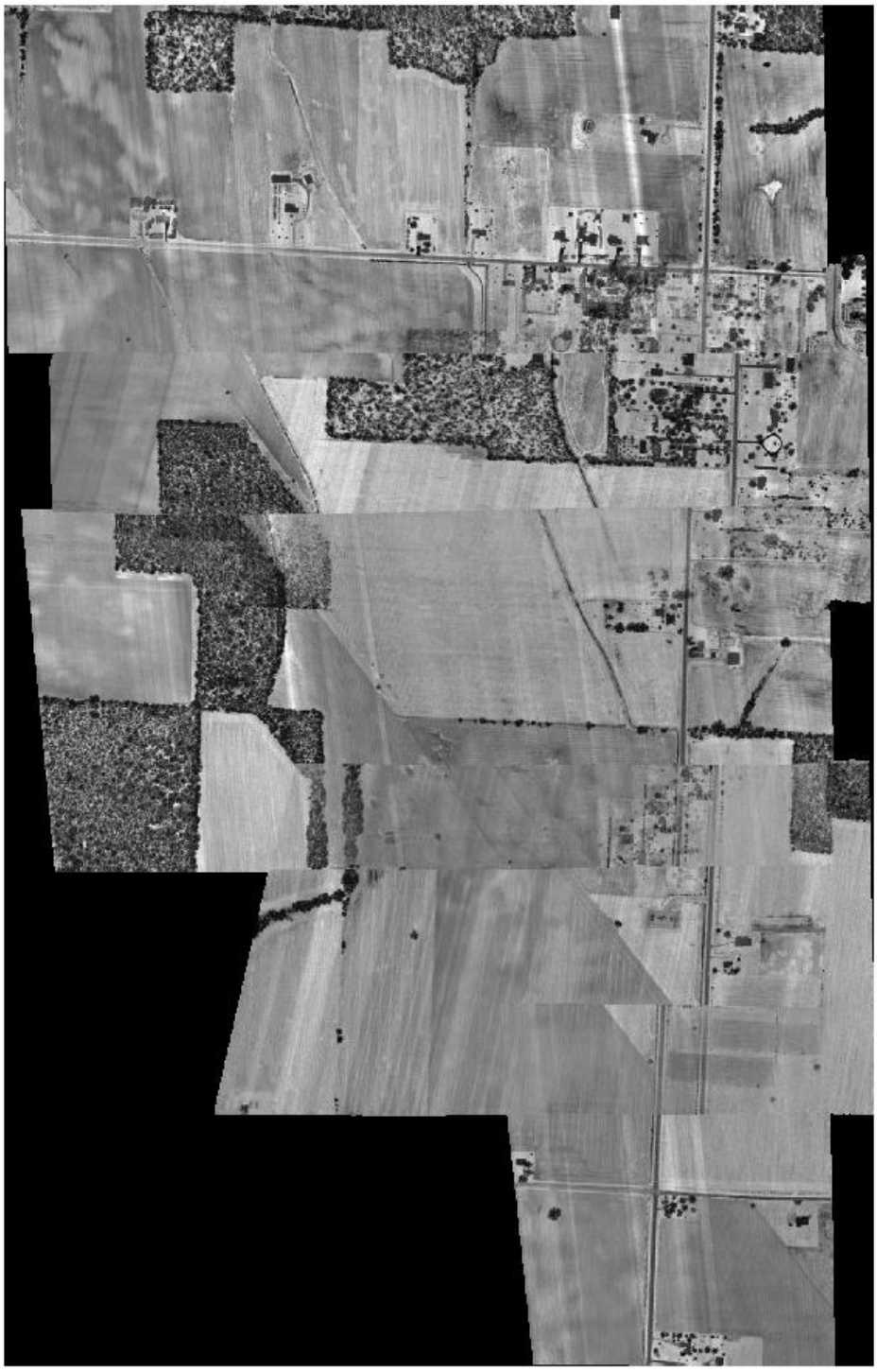


Figure 4-10 Simulated Biased-Added Vertical 1 under BFB Strip Configuration

Figure 4-11 Systematic-Bias-Model-Based Strip Mosaicing Result under BFB Strip Configuration

To evaluate the performance of each test, absolute errors between simulated parameters and estimated parameters from each test is used. The error is calculated in terms of absolute distance of the strip discrepancies. In other words, the error is equal to the difference between the estimated and simulated parameters converted to pixels.

The error between the estimated values $\{\delta\Delta\hat{k}, d\hat{X}, d\hat{Y}\}$ and the simulated values $\{\delta\Delta k, dX, dY\}$ in the BFB strip configuration can be calculated by using equation (66), (67) and (68), respectively.

$$error = \left|\left(x'\delta\Delta\hat{k} - x'\delta\Delta k\right)/res\right| \tag{66}$$

$$error = \left|\left(d\hat{X} - dX\right)/res\right| = \left|\left(d\hat{X} - (\delta\Delta X - H\delta\Delta\varphi)\right)/res\right| \tag{67}$$

$$error = \left|\left(d\hat{Y} - dY\right)/res\right| = \left|\left(d\hat{Y} - (\delta\Delta Y + H\delta\Delta\omega)\right)/res\right| \tag{68}$$

Where,

x' is the maximum value of the lateral distance between the LiDAR point in and the projection of the flight trajectory onto the ground. Since the maximum strip width is 650m and the flight line is in the middle of each strip, the value of x' is equal to 325m.

res is the selected 1D interpolation resolution, whose unit is meter-per-pixel.

H is the flight height, which is a constant equal to 6500 feet.

$\delta\Delta X, \delta\Delta Y, \delta\Delta k, \delta\Delta\varphi$ and $\delta\Delta\omega$ are the selected bias parameters.

$d\hat{X}, d\hat{Y}$ and $\delta\Delta\hat{k}$ are the estimated parameters in BFB strip configuration.

The absolute error between the estimated values $\delta\Delta\hat{k}$ and the simulated values $\delta\Delta k$ in the FFF strip configuration can be calculated by using equation (66).

4.3.2.1 Effect of Biases in Lever-arm Offset and Boresight Angles

In this subsection, the effect of the different bias values in the lever-arm offset and boresight angle on the performance of every test is analyzed. Both FFF and BFB

strip configurations are considered. The interpolation resolution value is fixed at $0.7m/pixel$ because the nominal pulse spacing is equal to $0.7m$.

4.3.2.1.1 FFF Strip Configuration

The absolute error between $\delta\Delta\hat{k}$ and $\delta\Delta k$ under various lever-arm offset bias and boresight angle bias is shown below.

Table 4-1 Absolute Error between $\delta\Delta\hat{k}$ and $\delta\Delta k$ (FFF, $res = 0.7m/pixel$)

Absolute Error (pixel)					
		Lever-arm Offset Bias (m)			
		0.05	0.6	1.2	2
Boresight Angle Bias	50''	0.202	0.220	0.207	0.204
	5'	0.058	0.058	0.049	0.060
	30'	0.029	0.028	0.032	0.032

In Table 4-1, boresight angle bias is presented in terms of angle in seconds and minutes. In terms of discrepancies in ground distances, 50" corresponds to 0.13 pixel at the maximum lateral distance of 325m. The corresponding numbers for 5' and 30' are 0.86 pixel and 5.1 pixels, respectively. All errors are in the sub-pixel level, indicating that there are hardly any noticeable strip discrepancies in the final mosaic. This shows that the proposed algorithm can output high quality LiDAR strip mosaics

By examining the errors along each row of the table, we can conclude that the bias in lever-arm offset does not significantly affect the absolute error. As can be seen from equation (59) and (60) and Figure 4-12, the lever-arm offset biases cancel each other in the adjacent strips. On the contrary, the bias in boresight angle has significant influence on the absolute error as can be seen by examining the error along each column

of the table. We speculate that this is possibly because larger boresight angle bias introduces larger discrepancies between strips, which are easier detected as seen from equation (45) and (46) and Figure 4-13. As mentioned above, 50", 5' and 30' in boresight angle biases correspond to 0.13 pixel, 0.86 pixel and 5.1 pixels, respectively. For example, the discrepancy caused by the bias is equal to 0.13 pixels when the boresight angle bias is equal to 50". Since the discrepancy is in the sub-pixel level, it vanishes during the interpolation process, leading to large difference between the estimation and simulated value.

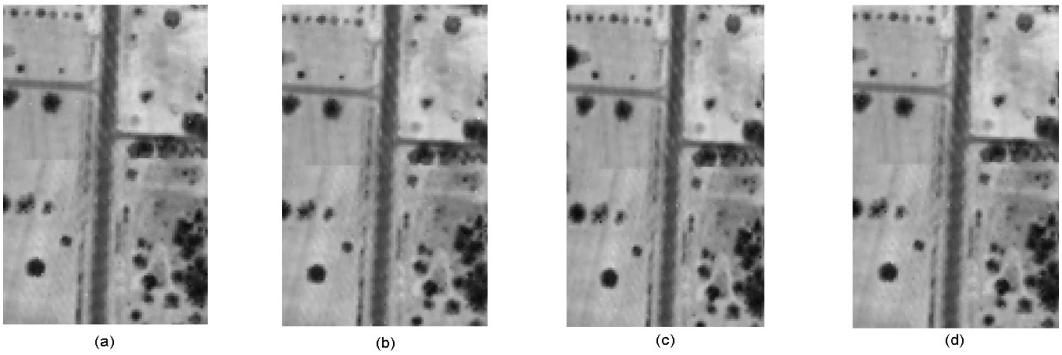


Figure 4-12 Same Enlarged Area Shows Lever-arm Offset Biases Cancellation for Different Bias Values (a) 0.05m (b) 0.6m (c) 1.2m and (d) 2.0m

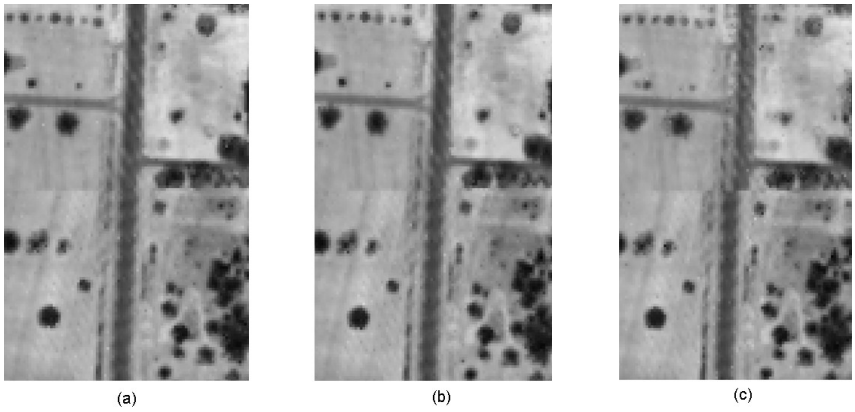


Figure 4-13 Same Enlarged Area Shows Boresight Angle Biases Influence for Different Bias Values (a) 50'' (b) 5' and (c) 30'

4.3.2.1.2 BFB Strip Configuration

The absolute error between $\delta\Delta\hat{k}$ and $\delta\Delta k$ under various lever-arm offset bias and boresight angle bias is shown below.

Table 4-2 Absolute Error between $\delta\Delta\hat{k}$ and $\delta\Delta k$ (BFB, $res = 0.7m/pixel$)

Absolute Error (pixel)					
		Lever-arm Offset Bias (m)			
		0.05	0.6	1.2	2
Boresight Angle Bias	50''	0.218	0.218	0.206	0.214
	5'	0.358	0.360	0.062	N/A
	30'	0.385	0.027	0.029	0.391

Unlike the FFF strip configuration, the effect of bias in level-arm offset and boresight angle bias appears to be random. However, the actual values of the errors for a range of boresight angle biases appears to be less than half a pixel and therefore invisible in a mosaic. Similar conclusions can be reached from Table 4-3 and 4-4 below.

The absolute error between $\mathrm{d}\hat{X}$ and dX under various lever-arm offset bias and boresight angle bias is shown below.

Table 4-3 Absolute Error between $\mathrm{d}\hat{X}$ and dX (BFB, $res = 0.7m/pixel$)

Absolute Error (pixel)					
		Lever-arm Offset Bias (m)			
		0.05	0.6	1.2	2
Boresight Angle Bias	50''	0.193	0.107	0.109	0.297
	5'	0.512	0.379	0.283	N/A
	30'	0.009	0.402	0.301	0.000

The absolute error between $\mathrm{d}\hat{Y}$ and dY under various lever-arm offset bias and boresight angle bias is shown below.

Table 4-4 Absolute Error between $\mathrm{d}\hat{Y}$ and dY (BFB, $res = 0.7m/pixel$)

Absolute Error (pixel)					
		Lever-arm Offset Bias (m)			
		0.05	0.6	1.2	2
Boresight Angle Bias	50''	-0.049	-0.120	0.019	0.463
	5'	-0.015	0.060	0.200	N/A
	30'	-0.213	-0.137	-0.140	-0.139

Chapter 5

Conclusions and Recommendation for Future Work

In this thesis, a technique to mosaic multiple LiDAR strips is developed and implemented. The technique is based on the LiDAR systematic biases model and takes advantages of ARFD-based LiDAR strip matching algorithm. The proposed method was successfully implemented to create LiDAR mosaics with matching errors of less than half a pixel, even when significant errors were synthetically introduced into the original strips. The ARFD based point matching algorithm together with the modified bundle adjustment method derived specifically for the modern LiDAR strips worked well to result in such minimal errors. The assumptions in the derivation of the modified bundle adjustment were also validated by the mosaicing results.

In the future, advanced version of the LiDAR strip mosaicing technique can be developed in order to deal with more types of systematic biases, like the bias in encoder angle and range measurement [2]. Besides, if the mosaicing technique is extended to 3D, which includes tasks like detecting 3D point correspondences, estimating the 3D rigid body transformation and determining the 3D discrepancies, a full LiDAR adjustment can be achieved.

References

[1] Shan, Jie, and Charles K Toth. “Chapter 1: Introduction to Laser Ranging, Profiling, and Scanning.” In *Topographic Laser Ranging and Scanning: Principles and Processing*. CRC Press, 2008

[2] Bang, Ki-In. “Alternative Methodologies for LiDAR System Calibration.” Doctoral Dissertation, University of Calgary, 2010.

[3] Sumerling, G. “Lidar Analysis in ArcGIS 10 for Forestry Applications.” *PDF White Paper. ESRI Inc* 1 (2011): 1–53.

[4] Gonzalez-Aguilera, Diego, Eugenia Crespo-Matellan, David Hernandez-Lopez, and Pablo Rodríguez-Gonzálvez. “Automated Urban Analysis Based on LiDAR-Derived Building Models.” *Geoscience and Remote Sensing, IEEE Transactions on* 51, no. 3 (2013): 1844–51.

[5] Liu, Xiaoye, Zhenyu Zhang, Jim Peterson, and Shobhit Chandra. “LiDAR-Derived High Quality Ground Control Information and DEM for Image Orthorectification.” *GeoInformatica* 11, no. 1 (2007): 37–53.

[6] Yao, Wei, Stefan Hinz, and Uwe Stilla. “Extraction and Motion Estimation of Vehicles in Single-Pass Airborne LiDAR Data towards Urban Traffic Analysis.” *ISPRS Journal of Photogrammetry and Remote Sensing* 66, no. 3 (2011): 260–71.

[7] Wehr, Aloysius . “Chapter 4: Lidar System and Calibrations.” In *Topographic Laser Ranging and Scanning: Principles and Processing*. CRC Press, 2008

[8] Wehr, Aloysius, and Uwe Lohr. “Airborne Laser Scanning—an Introduction and Overview.” *ISPRS Journal of Photogrammetry and Remote Sensing* 54, no. 2 (1999): 68–82.

[9] Lakshman, Mythreya Jayendra. “A New Feature Descriptor for LiDAR Strip Matching.” The University of Texas at Arlington, 2014

[10] Gatziolis, Demetrios, Hans-Erik Andersen, and others. “A Guide to LIDAR Data Acquisition and Processing for the Forests of the Pacific Northwest,” 2008

[11] Baltsavias, Emanuel P. “Airborne Laser Scanning: Existing Systems and Firms and Other Resources.” *ISPRS Journal of Photogrammetry and Remote Sensing* 54, no. 2 (1999): 164–98.

[12] Lemmens, Mathias. “Airborne Lidar Sensors.” *GIM International* 21, no. 2 (2007): 24–27.

[13] Habib, Ayman, Ki In Bang, Ana Paula Kersting, and Jacky Chow. “Alternative Methodologies for LiDAR System Calibration.” *Remote Sensing* 2, no. 3 (2010): 874–907.

[14] Kersting, Ana Paula, Ayman F Habib, Ki-In Bang, and Jan Skaloud. “Automated Approach for Rigorous Light Detection and Ranging System Calibration without Preprocessing and Strict Terrain Coverage Requirements.” Optical Engineering 51, no. 7 (2012): 076201–1.

[15] Toth, Charles K. “Chapter 8 Strip Adjustment and Registration.” *Topographic Laser Ranging and Scanning-Principles and Processing*, 2009, 235–68.

[16] Toth, Charles K, and Dorota A Grejner-Brzezinska. *Airborne LiDAR Reflective Linear Feature Extraction for Strip Adjustment and Horizontal Accuracy Determination*. Ohio State University, 2009.

[17] Lowe, David G. “Distinctive Image Features from Scale-Invariant Keypoints.” *International Journal of Computer Vision* 60, no. 2 (2004): 91–110.

[18] Mikolajczyk, Krystian, and Cordelia Schmid. “A Performance Evaluation of Local Descriptors.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27, no. 10 (2005): 1615–30.

[19] Bay, Herbert, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. “Speeded-up Robust Features (SURF).” *Computer Vision and Image Understanding* 110, no. 3 (2008): 346–59.

[20] Fischler, Martin A, and Robert C Bolles. “ Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.” *Communications of the ACM* 24, no. 6 (1981): 381–95.

[21] Derpanis, Konstantinos G. “Overview of the RANSAC Algorithm.” *York University* 68 (2010).

[22] Jain, PM, and Vijaya K Shandliya. “A Review Paper on Various Approaches for Image Mosaicing.” *International Journal of Computational Engineering Research* 3, no. 4 (2013).

[23] Moffitt, Francis H., and Edward M. Mikhail. *Photogrammetry*. 3rd ed. Harper & Row, Publishers, Inc, 1980.

[24] Fu, Zhaoxia, and Liming Wang. “Optimized Design of Automatic Image Mosaic.” *Multimedia Tools and Applications*, 2013, 1–12.

[25] Reddy, B Srinivasa, and Biswanath N Chatterji. “An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration.” *IEEE Transactions on Image Processing* 5, no. 8 (1996): 1266–71.

[26] Brown, Lisa Gottesfeld. “A Survey of Image Registration Techniques.” *ACM Computing Surveys (CSUR)* 24, no. 4 (1992): 325–76.

[27] Heitger, F., Rosenthaler, L., von der Heydt, R., Peterhans, E., and Kuebler, O. 1992. “Simulation of neural contour mechanism: From simple to end-stopped cells.” *Vision Research*, 32(5):963–981.

[28] Förstner, W. 1994. “A framework for low level feature extraction.” In Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden, pp. 383–394

[29] Sheng, Yongwei. “Quantifying the Size of a Lidar Footprint: A Set of Generalized Equations.” *Geoscience and Remote Sensing Letters, IEEE* 5, no. 3 (2008): 419–22.

[30] Zitova, Barbara, and Jan Flusser. “Image Registration Methods: A Survey.” *Image and Vision Computing* 21, no. 11 (2003): 977–1000.

[31] Szeliski, Richard. “Image Alignment and Stitching: A Tutorial.” *Foundations and Trends® in Computer Graphics and Vision* 2, no. 1 (2006): 1–104.

[32] Schmid, Cordelia, Roger Mohr, and Christian Bauckhage. “Evaluation of Interest Point Detectors.” *International Journal of Computer Vision* 37, no. 2 (2000): 151–72.

[33] Harris, Chris, and Mike Stephens. “A Combined Corner and Edge Detector.” In *Alvey Vision Conference*, 15:50. Manchester, UK, 1988.

[34] Beis, Jeffrey S, and David G Lowe. “ Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces.” In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1000–1006. IEEE, 1997.

[35] Brown, Matthew, and David G Lowe. “Automatic Panoramic Image Stitching Using Invariant Features.” *International Journal of Computer Vision* 74, no. 1 (2007): 59–73.

[36] McLauchlan, Philip F, and Allan Jaenicke. “Image Mosaicing Using Sequential Bundle Adjustment.” *Image and Vision Computing* 20, no. 9 (2002): 751–59.

[37] Fang, Xianyong, Bin Luo, Haifeng Zhao, and Yiwen Zhang. “A 5-Parameter Bundle Adjustment Method for Image Mosaic.” In *Computer and Information Science, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on*, 1063–67. IEEE, 2009.

[38] Triggs, Bill, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. “Bundle Adjustment—a Modern Synthesis.” In *Vision Algorithms: Theory and Practice*, 298–372. Springer, 2000.

[39] Nielsen, Allan Aasbjerg. “Least Squares Adjustment: Linear and Nonlinear Weighted Regression Analysis.” *Danish National Space Center/Informatics and Mathematical Modelling, Technical Univ. of Denmark*, 2007.

[40] Monnin, David, Etienne Bieber, Gwenaél Schmitt, and Armin Schneider. “An Effective Rigidity Constraint for Improving RANSAC in Homography Estimation.” In *Advanced Concepts for Intelligent Vision Systems*, 203–14. Springer, 2010.

[41] Horaud, Radu, Francoise Veillon, and Thomas Skordas. “Finding Geometric and Relational Structures in an Image.” In *Computer Vision—ECCV 90*, 374–84. Springer, 1990.

[42] Chum, Ondrej, and Jiri Matas. “ Matching with PROSAC-Progressive Sample Consensus.” In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:220–26. IEEE, 2005.

[43] Hartley, Richard, and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003.

[44] Cottier, J.C. 1994. “Extraction et appariements robustes des points d’intérèt de deux images non étalonnées”. Technical Report, LIFIA–IMAG–INRIA Rhône-Alpes

Biographical Information

Yanyan Hu received his B.S.E.E and M.S.E.E from University of Texas at Arlington (UTA) in 2012 and 2014, respectively. From 2010 to 2011, he worked with the Resonant Sensor Inc. as an undergraduate research assistant on hardware optimization and upgrading. From 2012 to 2014, he worked with the Virtual Environment Laboratory as a research student. His research interests include signal/image processing applications, pattern recognition and computer vision.