5-2021

Algorithms for the Calibration and Correction of Texel Images Using Inertial Measurement Updates

Trevor C. Welch
Utah State University

ALGORITHMS FOR THE CALIBRATION AND CORRECTION OF TEXEL IMAGES USING INERTIAL MEASUREMENT UPDATES

by

Trevor C. Welch

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

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Scott E. Budge, Ph.D.
Major Professor

Calvin Coopmans, Ph.D.
Committee Member

Randall Christensen, Ph.D.
Committee Member

D. Richard Cutler, Ph.D.
Interim Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2021

ABSTRACT

Algorithms for the calibration and correction of texel images using inertial measurement updates

by

Trevor C. Welch, Master of Science

Utah State University, 2021

Major Professor: Scott E. Budge, Ph.D.
Department: Electrical and Computer Engineering

Scientific sensors can be used to create textured digital surface models (TDSM) that show both the spatial measurements of a surface and the texture associated with it. A TDSM can be created using a combination of data from a light detection and ranging (lidar) sensor, an inertial navigation system (INS), and a digital camera. This thesis presents the development of such a sensor using low-cost, commercial off the shelf components (COTS) called a Texel Camera (TC). The TC acquires lidar data, which creates a point cloud describing the surface of the earth and combines that with position and attitude data from the INS and textures that point cloud with a digital image.

The development process of the TC includes creating a mechanical structure, software architecture to support data acquisition, calibration of the sensor, and algorithms developed to correct for errors. The structure must contain all of the vital components and meet size constraints for a small, unmanned aerial system (sUAS) that the TC will be flown on. The calibration process involves alignment of the sensors and mapping pixels to 3D points on a normalized lidar coordinate space. Motion correction methods are developed using the understanding of the rotational lidar sensor and the motion of the sUAS during flight.

The TC packages its three main sensors as well as on-board storage devices and processors to capture data for the create of TDSMs. The methods to develop the TC and the motivation to create correction algorithms are presented in this thesis as well as the findings.

(83 pages)

PUBLIC ABSTRACT

Algorithms for the calibration and correction of texel images using inertial measurement updates

Trevor C. Welch

A 3D terrain map can be developed to survey the earth's surface and provide scientists with a method of accurately modeling terrain. Some of the applications include agricultural surveys, disaster management, and object classification. In order to develop a 3D terrain map, a sensor must capture all of the data to create the map. This 3D mapping tool can be made by combining a sensor that take samples of points on the earth's surface and combines that with a digital camera that takes an image and lays it across those samples of the earth.

In order to take data over a large area, the entire system is flown on a small drone and must be small enough to allow the drone to fly for several minutes. While the drone is flying, a GPS sensor tracks the system's location and orientation while taking data so they can be correlated with the 3D terrain map. Utah State University has developed a sensor for this application, called a texel camera, which uses laser measurements, digital images, and a GPS system to develop 3D terrain maps. The texel camera includes methods for correcting the terrain maps using the physics of the sensors and the motion of the drone during flight.

ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Scott Budge, for giving me the opportunity to work on this project with him and for his continued guidance through my education. I also want to acknowledge Dr. Calvin Coopmans and Dr. Randy Christensen for their help in developing the texel camera and the motion correction algorithms. I would also like to thank my supportive parents who have helped me through my education and allowed me to continue to do what I love. I am so grateful for my girlfriend, Madison Ferguson, who has helped me stay dedicated and focused through my education and pushes me every day.

I am thankful for all of my fellow students at the Center for Advanced Imaging Ladar, specifically Bikalpa Khatiwada for his help during this project. Also, the students at the AggieAir Research Center: Colby Jones, Richard Loveland, Kevin Plaizier, and Ethan Payne. I am thankful for the staff in the ECE department including Dr. Todd Moon, Dr. Jake Gunther, Heidi Harper, and Tricia Brandenburg.

I acknowledge this research has been funded by the Utah Science Technology and Research (USTAR) program.

Trevor Welch

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ACRONYMS

CAIL	Center for Advanced Imaging LADAR
COP	center of projection
COTS	commercial off the shelf sensor
ENU	east north up
FOV	field of view
HFOV	horizontal field of view
INS	inertial navigation system
LOS	line of sight
SSD	solid state drive
sUAS	small unmanned aerial system
SWaP	size weight and power
TC	texel camera
TDSM	textured digital surface map
TRL	technology readiness level
UAV	unmanned aerial vehicle
VFOV	vertical field of view
WGS	world geodetic system

CHAPTER 1

INTRODUCTION

The development of sensors using light detection and ranging (lidar) for small unmanned aerial system (sUAS) has become a topic of interest in various research areas, including terrain mapping [1], vegetation and crop management [2], object classification [3], among others. Aerial mapping provides valuable information for users and creates complex challenges when designing sensing systems. As advancements in technology enable lidar sensors to be more available than ever before, it becomes easier to develop integrated lidar systems for remote sensing applications. Including lidar, the availability of commercial off-the-shelf (COTS) technology for drone mapping has driven innovation and creation of new systems. The improvements in lidar technology create the opportunity to develop cost-effective and lightweight systems for small, unmanned aerial system (sUAS) applications. Lidar is one type of sensor used in aerial applications, and by adding multiple sensors, such as electro-optical (EO) cameras or inertial navigation systems (INS), a more robust system can provide complex data sets.

This thesis focuses on the integration of a lidar sensor, digital camera, inertial navigation system, and processor to create a sensor package for the acquisition of sensor data and the creation of textured digital surface models (TDSM). With collaboration from the Center for Advanced Imaging Ladar (CAIL) [4] and the AggieAir Research Group at Utah State University [5], a sensor package has been developed for the acquisition of data used to create TDSMs. The sensor package has been tested in various environments, including an indoor laboratory with a static sensor, an outdoor lab environment with a dynamic sensor, and an outdoor experimental environment (completed drone flight) with a dynamic sensor. The results from all of these experiments are included in this document.

As mentioned, the sensor package is composed of a lidar sensor, digital camera, inertial navigation system, and on-board processor. The system is referred to as a texel camera

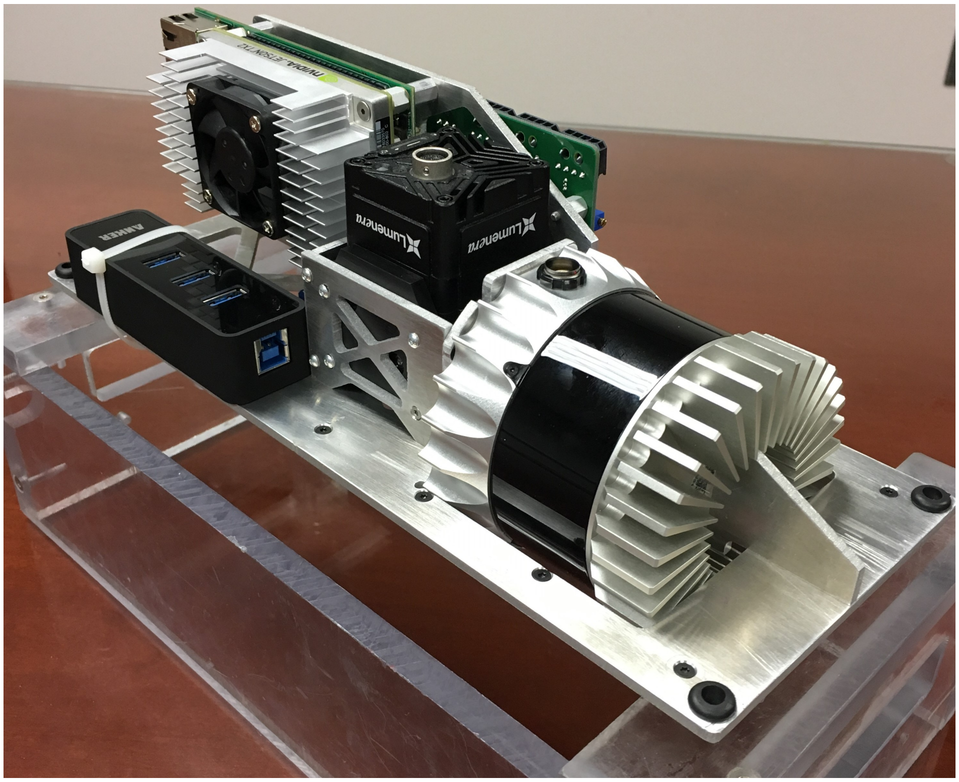


Fig. 1.1: Assembled texel camera.

(TC) and is designed to be compatible with existing fixed-wing and rotorcraft sUAS. The constructed TC can be seen in Fig. 1.1.

1.1 Objectives

The continuous increase in lidar performance, range, and accuracy provides obvious benefits for sensors such as the TC and allows sensors for many applications to be developed using relatively inexpensive lidar. This project was built on the goal of developing scientifically accurate TDSMs using low-cost COTS sensors. The objectives that developed to complete that goal involved both hardware and software milestones including:

1. Mechanically construct and align the sensor package for use in an sUAS.

2. Develop a calibration method for the TC in a static environment to map image pixels to lidar shots.

3. Develop an algorithm to compensate for disturbances experienced by the sUAS during flight and compute normalized lidar coordinates using the correction.

All of the objectives listed above will be described in detail in their respective sections in this document.

1.2 Overview

This work describes the development of the TC, including physical design considerations and sensor calibration, as well as detail the algorithms developed to correct errors in the captured data due to changes in position and orientation during flight.

Chapter 2 is a literature review involving lidar, aerial mapping, and similar topics discussed in this paper to provide background knowledge. Chapter 3 provides an overview of the TC, describes the development of the TC mechanical structure, and describes the data acquisition process. Chapter 4 describes the static calibration of the TC, sensor alignment, and pixel-to-shot mapping. A method for dynamic calibration is covered in Chapter 5 that describes the development of an algorithm to compensate for translational and rotational motion during data acquisition. Lastly, Chapter 6 will present the findings of the processes seen in this document and describe future work.

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND

The use cases for lidar and digital imagery are obvious in scenarios including obstacle avoidance [6], object detection [7], scene reconstruction [8], aerial mapping [9], as well as others [10–13]. Recently, lidar efforts have been driven by the autonomous vehicle industry, pushing research to improve intelligent vehicles in all areas [14, 15]. While lidar allows complex analysis of shape, structure, reflectivity, and range, a digital camera provides important texture and color data to describe features.

To better understand the TC's efforts, a review of state-of-the-art in lidar and digital imagery in remote sensing applications is discussed in this chapter. As a note, not all topics discussed in this section will involve sUAS specific applications, but prove to be useful examples of the previous and current work involving lidar related applications.

2.1 Fusion and Registration of Images and Lidar Data

When creating 3D models, such as TDSMs, the data relies on the accurate registration of non-textured lidar data to image pixels. The fusing efforts involved in this work were previously completed by CAIL using automatic registration techniques [16,17]. An example of a registered texel image captured using the TC can be seen in Fig. 2.1.

Another fusing method is described by Wang [18] that is also an automatic registration method using novel features called 3CS, composed of line segments created by matching features between aerial images and lidar data. The 3CS features are then matched between datasets for the fusion of data. Berrio [19] discusses the implementation of a convolutional neural network as a classifier of the environment in automated vehicle applications and uses semantic information between lidar and image data to fuse the sensor information.

2.2 Calibration of Lidar and Image Sensors

A sensor package designed to capture lidar and digital imagery is not a new development, but calibration techniques are still being developed to increase the accuracy of fused data sets. Proper calibration is critical to the accuracy of data and is a crucial design requirement of the TC and other optical sensors. The calibration process can often become complicated and involved, and previous studies have approached the same problem in different ways.

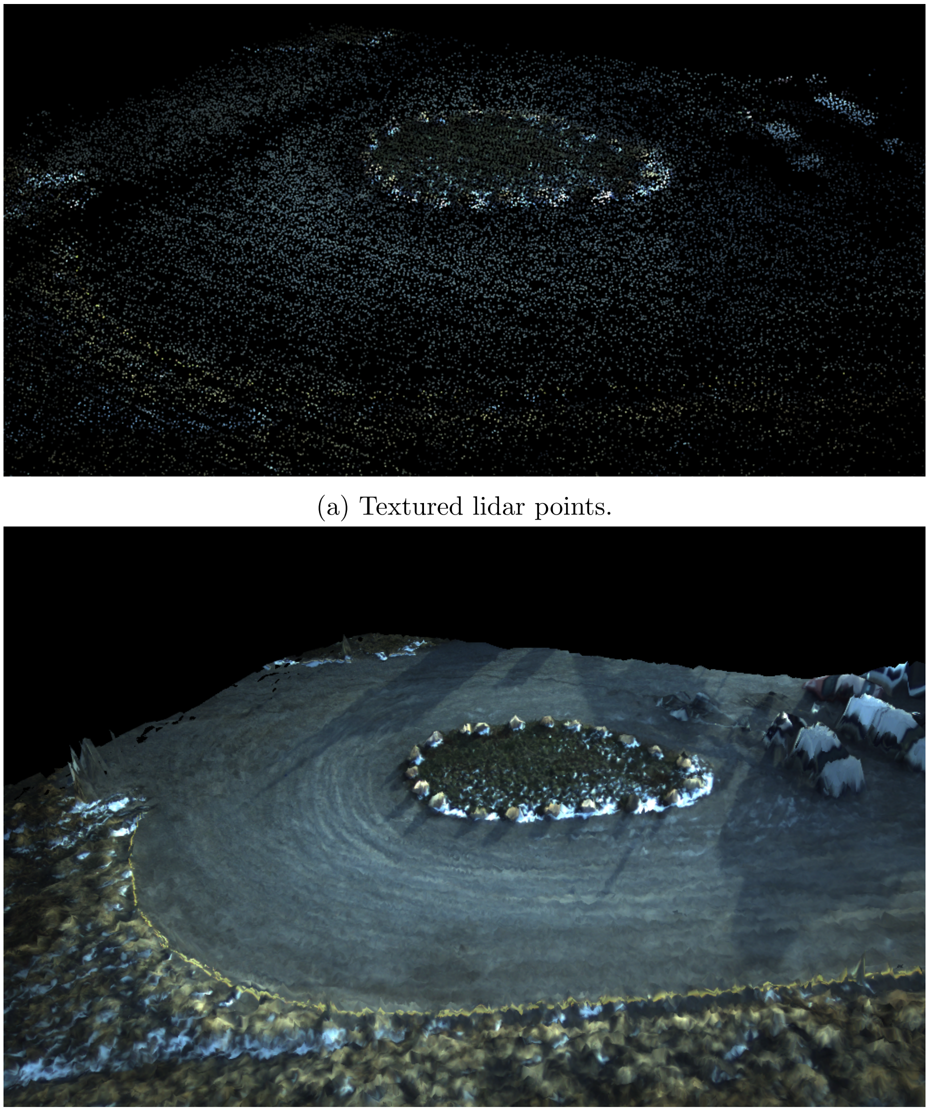
In many calibration examples using a camera and ranging sensor, a planar calibration target is used [20–22]. A common calibration target is a checkerboard pattern with evenly sized squares placed on a planar surface. A checkerboard calibration target will also be used for the TC but the method for calibrating the camera to the lidar differs from other methods.

There have been efforts to develop an autocalibration method for lidar and digital imagery by using edge alignment [23]. The approach does not require known feature detection, but instead fuses the data sets and evaluates edge alignment quality to estimate the calibration parameters. The method does not have known calibration parameters before the autocalibration, which differs from the process proposed for the TC. The technique discussed in the autocalibration uses a camera and lidar with different sampling information that could increase calibration errors but suggests that the fusion of the data to estimate calibration parameters offsets this difficulty. Various other automatic calibration methods have been studied that estimate camera parameters and provide camera-to-range calibration [24–26].

Moghadam [27] presents an extrinsic calibration method between range and image sensors that do not require planar calibration patterns. The article mentions approaches that use calibration patterns are limited because the scene must be modified with artificial targets. Instead of artificial targets, the algorithms rely on 3D lines extracted from the lidar point cloud and 2D lines from the image along with an optimization algorithm to estimate a relative translation and rotation between coordinate frames.

Another method explains pixel and 3D point alignment by extracting corresponding lidar points and image pixels, then calculating a linear alignment matrix, without considering the distortion parameters of the camera first [15]. The linear alignment matrix is then optimized using maximum likelihood estimation to examine the camera parameters. The document states that estimating a camera's intrinsic parameters before the fusion of data accumulates errors that reduce accuracy compared to state-of-the-art methods.

The method proposed for the TC differs from the practices described above by the alignment of the lidar and camera, consideration of the camera's distortion parameters, and a motion correction algorithm to compensate for movement during a lidar scan. The methods discussed above do not take into account a vital alignment factor, parallax, which is the physical offset between the sensors that cause errors in the calibration of the data sets. There are significant differences in the methods previously developed and the method presented in this thesis.



(a) Textured lidar points.

(b) Registered texel image.

Fig. 2.1: Textured lidar points and registered texel image.

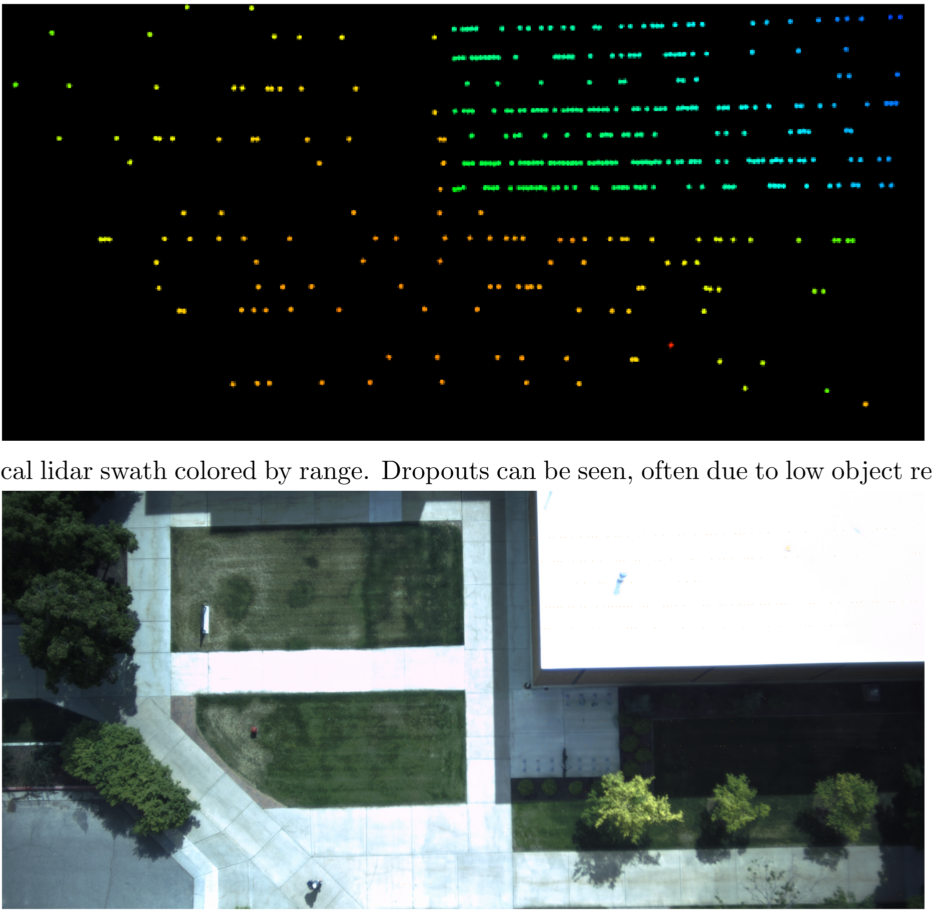
CHAPTER 3
TEXEL CAMERA DEVELOPMENT

3.1 Texel Camera Overview and Definitions

As mentioned in Chapter 1, CAIL has developed a texel camera for aerial mapping applications for the creation of TDSMs. The TC described in this document is in its first research development to be used as an airborne sensor, but has had previous generations used as a handheld camera [28]. The purpose of the TC is to collect time-tagged and synchronous data from three sensors to create texel images and to integrate with an sUAS. Three sensors acquire data, and one central processing and graphical processing unit (CPU/GPU) is used to process and store data. These components include a lidar, camera, INS, and CPU/GPU, all of which will be discussed in detail in the following sections.

The data from the TC is captured in swaths, composed of a digital image where the cross-track (horizontal) dimension is larger than the in-track (vertical) dimension, a cross-track sweep of lidar measurements (shots), and multiple measurements of the position and attitude (pose) of the TC at the time the swath was acquired. An example of a single scan of lidar measurements and the image associated with the scan can be seen in Fig. 3.1. Lidar data has additional attributes besides range, including azimuth, elevation, and reflectivity, some of which are used during data fusion to improve accuracy.

The TC acquires image and lidar data at a rate of 10 Hz during operation and flies at an altitude between 60 and 90 m. During a lidar scan, the INS captures position and attitude data, and one image is taken. The combination of this data is called a texel swath or frame. The data is post-processed to combine all three datasets using timestamp information and calibration parameters to create a texel image for each swath. This post-processing does not require registration estimation.



(a) A typical lidar swath colored by range. Dropouts can be seen, often due to low object reflectivity.

(b) An image associated with the lidar scan in (a).

Fig. 3.1: A standard lidar scan and the associated image captured and time-tagged to the lidar scan shown in (a).

3.1.1 Texel Camera Specifications

Before the development of the TC, specifications and requirements were written for each sensor to maximize the system's performance. A thorough market analysis was done for more than 40 sensors to ensure all sensors would meet requirements, the most notable being size, weight, and power (SWaP). A table listing the specifications of the TC can be seen in Table 3.1. Consideration must be taken when comparing lidar specifications and end-user requirements, due to the variance in performance of low-cost lidar systems compared to their specifications [29].

The lidar sensor in Fig. 3.2a is a 16 channel, 360 degree, rotational lidar configured to

Requirement	Specification	Units
Minimum Range	60 @ 0.3 refl.	m
Digital Image Width	$\geq$ 2048	pixels
Swath Horizontal FOV	$\geq$ 30	degrees
Swath Rate	$\geq$ 10	swaths/s
Scan Time	$\leq$ 15	ms
Position Knowledge	$<$ 3	m
Attitude Knowledge	$<$ 0.5	degrees
Power	$<$ 35	W
Dimensions (LxWxH)	26.67 x 12.7 x 8.73	cm
Mass	$<$ 1.9	kg

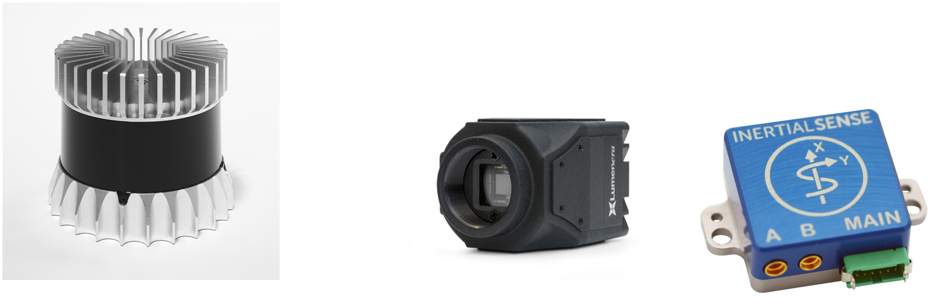
Table 3.1: Texel Camera Specifications

have a rotation rate of 10 Hz. There are 2048 calibrated columns with an angular sampling accuracy in the vertical and horizontal direction of less than 0.01 degrees. The vertical field of view (VFOV) of the lidar is 15.8 degrees, and the horizontal field of view (HFOV) is configurable and is often set between 30 and 40 degrees. Images from the camera are captured at a rate of 10 Hz with exposure times ranging from 0.5-1 ms depending on light conditions. The horizontal and vertical FOV of the camera is automatically calculated based on the lidar parameters (see Appendix A.1), but the maximum camera resolution in pixels is 2832 in the y (vertical) direction and 4240 in the x (horizontal) direction.

The INS is a micro-electro-mechanical system (MEMS), which measures pose (position and attitude) data while the TC is capturing range data. The pose information is used to determine the motion of the aircraft during the capture time of one swath. The pose data is used during post-processing to correct for motion, which is discussed more in Chapter 5. The processor used for the TC is the NVIDIA Jetson TX2 CPU/GPU and is used to run software to acquire and write data to an onboard solid-state drive (SSD). The individual TC components can be seen in Fig. 3.2.

3.2 Texel Camera Development

The development of the TC was divided into milestones to track the progress of the project. The first milestone involved the creation of a preliminary design seen in Fig. 3.3 and an in-depth market survey of COTS lidar, camera, processor, and INS sensors to ensure all



(a) Ouster 16 channel lidar

(b) Lumenera camera - lt1265rc

(c) Inertial Sense uINS

Fig. 3.2: The three main TC sensors, lidar, digital camera, INS.

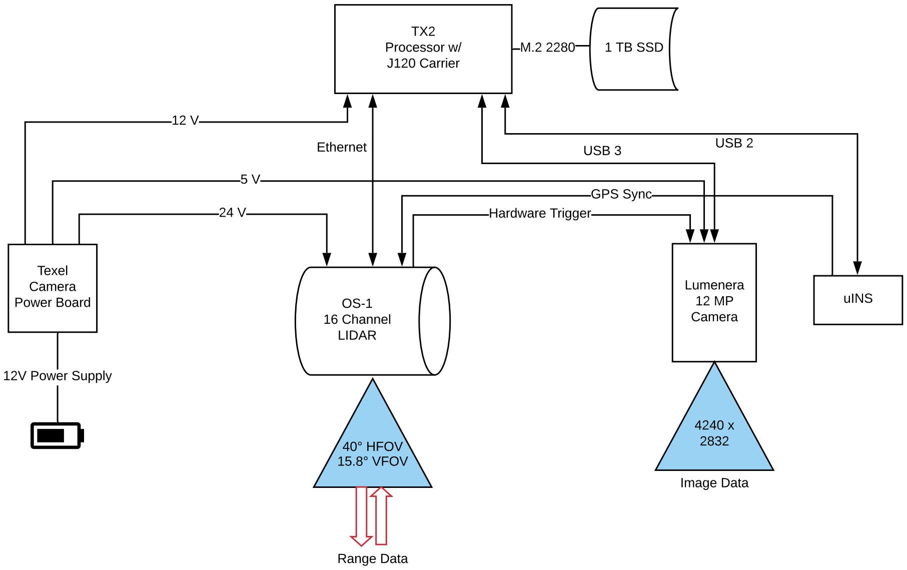


Fig. 3.3: High-level texel camera system diagram.

components would meet the design requirements of the TC. Next, the software was written to timestamp and combine the data streams of each sensor. Then a mechanical structure was designed to align the sensors on a rigid body mounted to fit the sUAS. Lastly, the TC was mounted and flown on mapping missions with the sUAS to collect data and develop TDSMs.

3.2.1 Coordinate Frames

The coordinate systems for the TC are defined (see Fig. 3.4) to provide information about the world coordinate system used and the pose of each sensor with reference to a world coordinate system. In this work, the world coordinate system is an east-north-up (ENU) system with the zero location being at the first measurement recorded from the INS. The world coordinate system will often be referred to as the inertial coordinate frame, or just the inertial frame, and is denoted with the script ENU.

Next is the frame of the INS, which will be called the TC's body frame, with its origin at the center of the INS unit and its axes in the north-east-down (NED) directions corresponding to x, y, and z, respectively. The body frame is denoted with the script b. The lidar coordinate frame will be denoted with the script L and has its origin at the center of the lidar sensor and follows an ENU convention, where east is the cross-track direction and north is the in-track direction. It is important to distinguish the pose of a lidar shot in the lidar sensor frame taken at a time t_1 from a lidar shot in the same frame taken at a time t_2. Equation (3.1) describes a vector $\boldsymbol{v}^b$ in the body frame being rotated to the lidar frame at the time t_1 and then being rotated to the inertial frame.

$$R_L^{ENU}(t_1)R_b^L(t_1)\boldsymbol{v}^b \tag{3.1}$$

3.2.2 Mechanical Structure

After selecting components for the TC, a structure was designed for mounting the hardware. The most important design considerations include the rigidness of the structure, instrumentation alignment, size, and weight. Critical design considerations had to be made for each sensor's placement on the structure to allow for the following assumptions:

1. All three sensors are aligned about the x-axis of the body frame.
2. The structure is rigid so that vibration and disturbances during flight do not affect the calibration of the lidar and camera.
3. There is negligible rotation about the z-axis (yaw) between the lidar, camera, and INS.

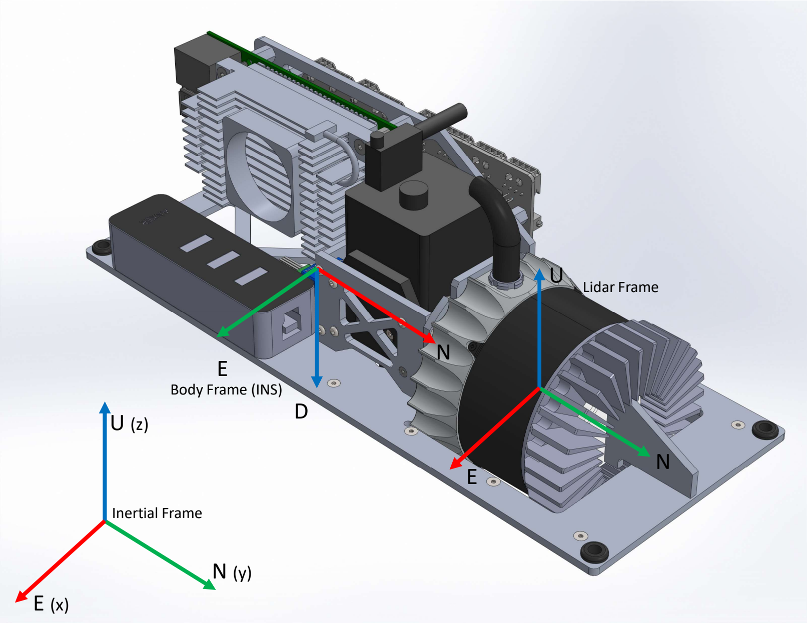


Fig. 3.4: TC coordinate frames: inertial frame, lidar frame, and body (INS) frame. The red axis is x, green axis is y, and the blue axis is z.

The first and third considerations put the center of the inertial unit inline with the center of the camera's projection and inline with the center of the lidar sensor. It may be true that there is a small physical offset in the y-direction between the sensors, but the structure is designed to make this offset small and is considered negligible in this document. This assumption becomes important when calibrating the camera and lidar and also for the equations developed to correct for motion.

The second assumption, a rigid structure, is accounted for by using aluminum stiffeners that run along the x-axis of the body frame on both sides of the bottom of the payload structure. The stiffeners provide support for any bending that may alter the lidar and camera center of projection's (COP) alignment.

All size and weight constraints allow the TC to be flown in the AggieAir fixed-wing and rotorcraft systems, as seen in Figures 3.5 and 3.6. Although the design was built for these aircraft, the system is light-weight and compact so that it can be integrated into almost any existing sUAS that can carry a payload less than 2 kg. This allows the TC to be a versatile platform that can be used for a variety of imaging scenarios.

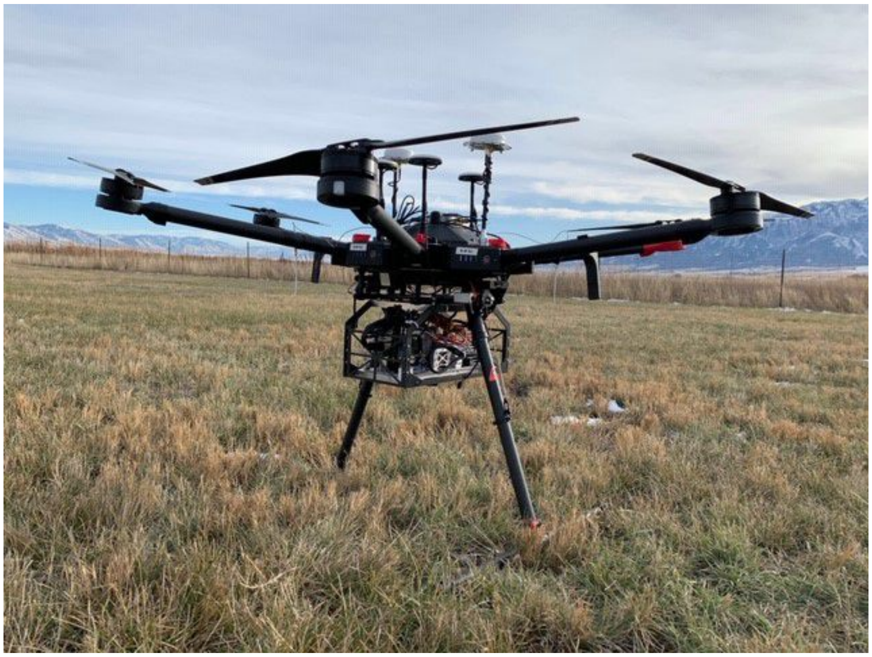


Fig. 3.5: Texel camera mounted in the DJI rotorcraft.

3.3 Texel Camera Operation

The TC relies on software and hardware for a variety of tasks in order to collect the data needed to develop TDSMs. The two key factors of operation include:

1. Time synchronization

2. Data acquisition

3.3.1 Time Synchronization

All three sensors (lidar, camera, and INS) have independent internal clocks that must be synchronized to one absolute time to register data collected from one sensor to another. Without any form of synchronization, the clocks experience drift relative to each other, and after approximately 1000 swaths, they can be off by more than 1 second. As the data acquisition time increases, so do the inconsistencies between clocks, so to fix this issue, synchronization is done using hardware and software to compensate for clock drift between sensors.

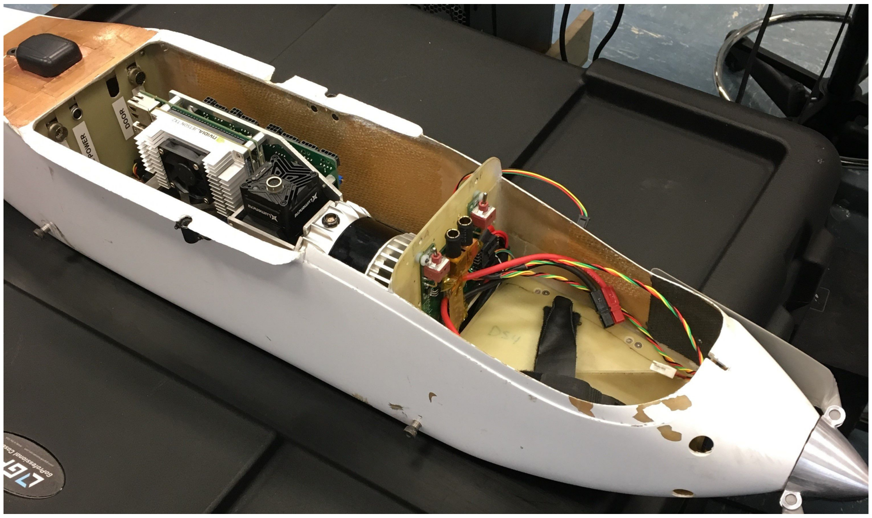


Fig. 3.6: Texel camera mounted in the AggieAir fixed-wing ”minion” aircraft.

The hardware component designed to synchronize each sensor is the power and routing board called the IO board. The IO board consists of DC to DC power converters, buffer circuits, and routing for crucial timing signals. An image of the IO board can be seen in Fig. 3.7.

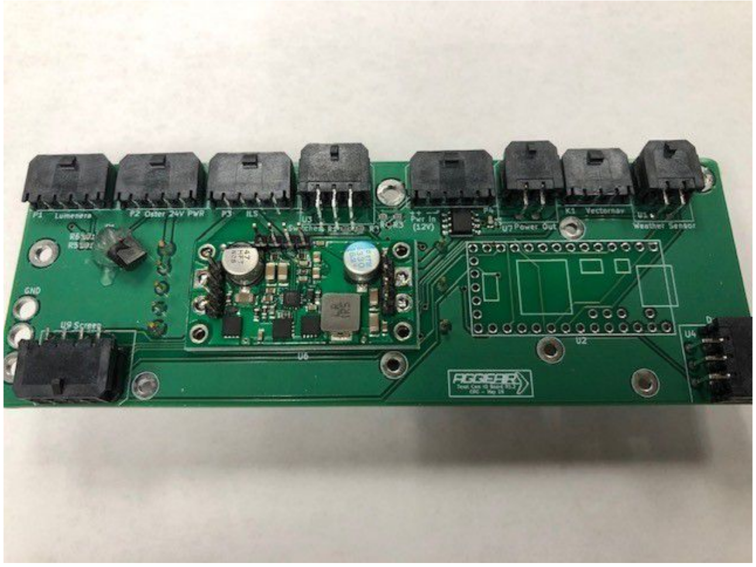


Fig. 3.7: Texel camera IO board.

The IO board includes a connection between the INS and the lidar sensor that synchronizes the lidar clock with the INS GPS pulse per second (PPS) signal. Since the PPS is received by the INS once every second from a GPS satellite, this synchronizes the lidar clock so there is essentially no drift, even after thousands of swaths. The PPS signal is output from the INS at 2.5 V with a 100 ms active high signal and a frequency of 1 Hz. This output signal is converted to 5 V using the simple circuit shown in Fig. 3.8. When the lidar sensor receives the signal, it sets all of the sub-second clock registers to zero, synchronizing the sensors every second to provide a constant time offset. While the TC is acquiring data, the software calculates the initial timestamp offset by reading the lidar clock and INS clock and this constant offset can be used for the entire data set to convert between the two times.

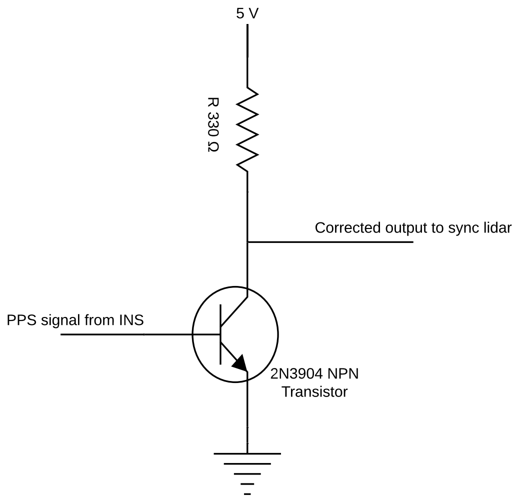


Fig. 3.8: Buffer circuit which corrects the output waveform to the lidar sensor for synchronization.

3.3.2 Data Acquisition

As mentioned previously, the data acquired by the TC consists of a digital image, a sweep of lidar shots in the cross-track direction, and pose measurements from the INS. The timing of these events is crucial to combine the data sets and with the timing synchronization discussed in Section 3.3.1 it is possible to get data in a specific order at accurate times. To describe the order that these events occur, some prior knowledge of the sensors is required.

During data acquisition, only lidar measurements taken within a configured HFOV (Fig. 3.9) pointing in the negative z-direction of the lidar frame (pointing towards the earth) are stored. A range of 200 to 256 lidar columns is typical for a swath (approximately 100-128 on both sides of the column in the center of the HFOV, nominally 1024).

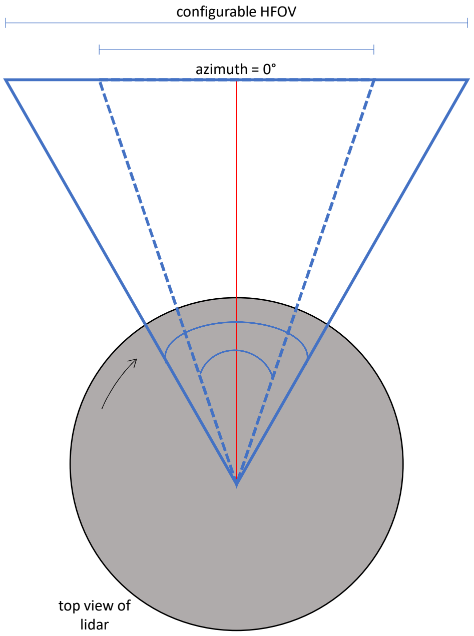


Fig. 3.9: Illustration of the lidar sensor with two different FOVs. When the lidar scan azimuth is within the configured HFOV, the TC acquires data.

Each lidar column captures 16 measurements simultaneously, but not all of the measurements are at the same azimuth angle. Although a lidar column isn't all in one line down a single azimuth, they are within a small azimuth range, as seen in Fig. 3.10. It is important to note that the measurements are captured at the same time, and this thesis will use the term, lidar column, to mean all measurements captured simultaneously and not specifically all at the same azimuth associated with a column.

When a lidar column enters the predefined HFOV, the measurements are stored as part of a swath and continue to be stored until the lidar column has no more measurements within the HFOV. During this time, one image is captured using a hardware trigger (see

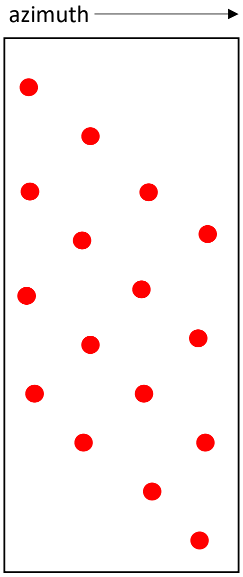


Fig. 3.10: Illustration of a lidar column with 16 lidar measurements taken at the same time, but not at the same azimuth.

Fig. 3.11). The hardware trigger pulse is output from the lidar sensor at the time of the 1024th lidar column and is used as an input to the digital camera, which triggers the camera exposure. The exposure happens within 47μsecs of the hardware pulse. This image is time-tagged and stored to be associated with the correct lidar scan during processing.

Pose measurements are recorded from the INS at various times throughout the lidar scan. The packets received from the INS hold the following attributes: time, quaternion body rotation with respect to NED, and WGS84 latitude, longitude, and altitude above ellipsoid in meters. The number of INS measurements per scan can vary from one to four and are used to calculate each lidar shots$x$, y, and z location in the world frame and are also used as described in Chapter 5 when correcting for motion. An illustration of the data acquisition events that occur in one swath can be seen in Fig. 3.11.

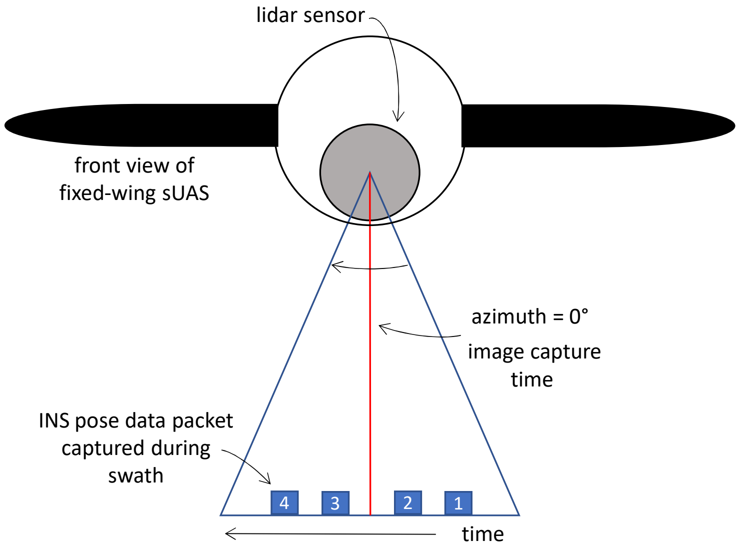


Fig. 3.11: The 1024th lidar column pointing in the negative z direction of the lidar frame at azimuth = 0°.

CHAPTER 4

TEXEL CAMERA STATIC CALIBRATION

The calibration of the texel camera is done in a static environment by using known features and targets to properly calibrate EO images and map lidar points to image pixels. There are three main steps involved in the static calibration process:

1. Calibrate the digital camera.
2. Align the principal rays of the camera and the lidar.
3. Estimate the lidar-to-camera mapping.

The completion of the static calibration is what creates a fully functional texel camera. For calibration, it is important to define the camera center of projection (COP) (seen in Fig. 4.1) along with the normalized image plane, the plane perpendicular to the principal ray of the camera at distance of one meter along the body frame z axis as seen in Fig. 4.1.

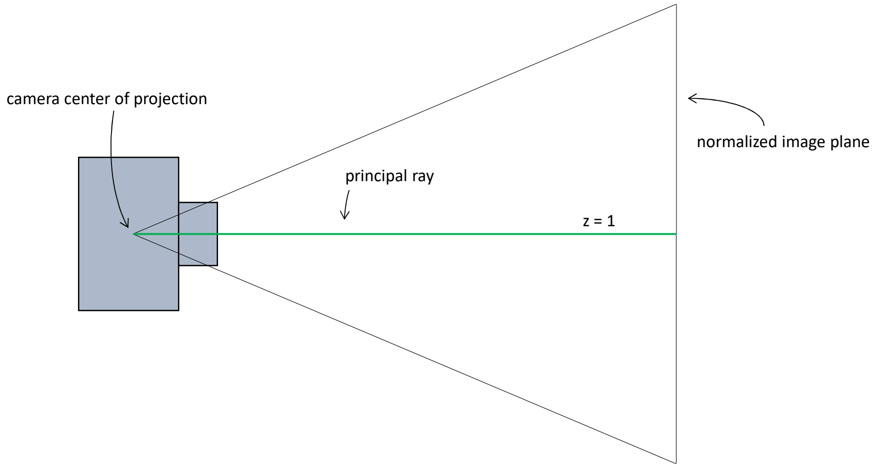


Fig. 4.1: Normalized image plane, perpendicular to the body frame z axis and at a distance of one meter.

4.1 Digital Camera Calibration

After the lidar and digital camera have been mounted to the payload structure, the sensors must be calibrated to combine the datasets and create TDSMs. Before the digital camera is calibrated, the image has distortions from the lens and imperfections during manufacturing. To correctly map lidar points to image pixels, the EO image must be properly calibrated. All cameras discussed in this document will follow the pinhole camera model. The pinhole model is shown in Fig. 4.2 with the image plane placed behind the camera COP, so the image is projected upside down. In this document, the camera model used is the pinhole model with the normalized image plane placed one meter in front of the camera center (see Fig. 4.1).

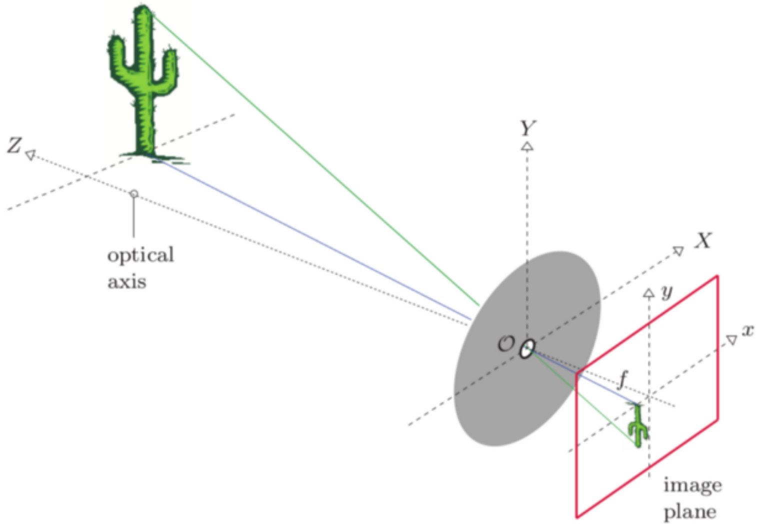


Fig. 4.2: Pinhole camera model. Where all light rays passing through the lens are assumed to pass through one point, the camera center.

Distortions in the image can be corrected by estimating the intrinsic parameters of the camera. The process of finding the intrinsic parameters uses a set of known points in the camera FOV and a known distance between those points to calculate the intrinsic parameters. In this case, the target is a black and white checkerboard with squares that are all the same size, as seen in Fig. 4.4. The calibration method used is a well-known method described in [30] by Heikkilä and Silvén. Using the model proposed by Heikkilä [30], the distorted pixels can be mapped to the normalized image plane $(x_n,\, y_n)$ using

$$\boldsymbol{p}_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix} = d_r \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \boldsymbol{d}_t, \tag{4.1}$$

where

$$d_r = 1 + k_1 r^2 + k_2 r^4 + k_5 r^6, \tag{4.2}$$

$$\boldsymbol{d}_t = \begin{bmatrix} 2k_3 x_n y_n + k_4(r^2 + 2x_n^2) \\ k_3(r^2 + 2y_n^2) + 2k_4 x_n y_n \end{bmatrix}, \tag{4.3}$$

and

$$r^2 = x_n^2 + y_n^2. \tag{4.4}$$

Equation (4.1) is inverted to get the normalized image values from the distorted image values. From these, the location of the point on the camera's sensor array in column-row pixel coordinates (c, r) on the digital image is given by

$$\begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} \tag{4.5}$$

where c is the pixel column, r is the pixel row, and $\boldsymbol{K}$ contains the camera's intrinsic parameters

$$\boldsymbol{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.6}$$

All of the parameters above $\begin{bmatrix} f_x & f_y & c_x & c_y & k_1 & k_2 & k_3 & k_4 & k_5 \end{bmatrix}$ are estimated using a MATLAB based calibration toolkit [31]. After this calibration was done for one of the cameras used in the TC, the camera calibration matrix was

$$\boldsymbol{K_b} = \begin{bmatrix} 5240.16323 & 0 & 2037.71488 \\ 0 & -5302.22580 & 1468.83678 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.7}$$

Common types of distortion are barrel distortion, where the image looks to be wrapped around a sphere, and pincushion distortion, where the image seems to be bowed inwards, as seen in Fig. 4.3. For the TC, the calibration process fixed minor barrel distortion, as seen in Fig. 4.4. Although the initial distortion was small, calibration is important to increase the accuracy of the sensor.

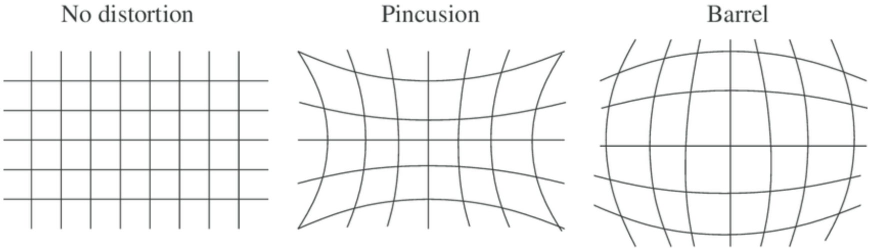
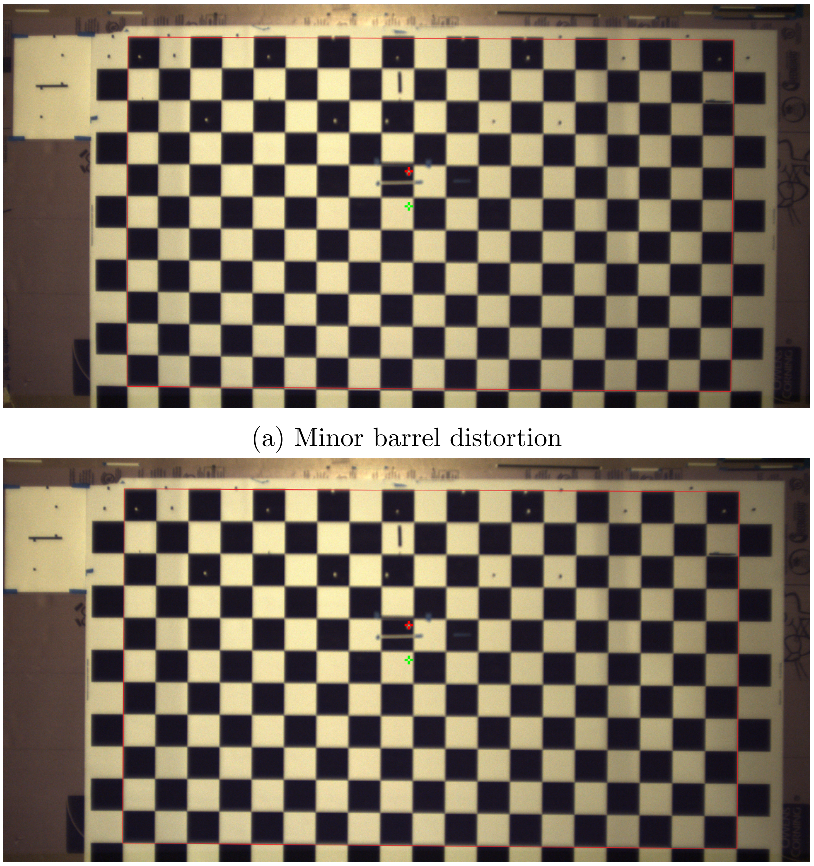


Fig. 4.3: The first image is the undistorted image after the effects of distortion have been removed. The image in the middle shows pincushion distortion, where the lines bow inward. The image on the right shows barrel distortion; the lines bend outward as if the image is being wrapped around a sphere.



(a) Minor barrel distortion

(b) Undistorted image

Fig. 4.4: Minor barrel distortion on the calibration target in (a) and (b) shows the same image after the distortions have been removed. The differences are small, but corrections as small as a few pixels create higher accuracy TDSMs. The green cross on an image represents the camera center and the red cross represents the lidar zero azimuth and zero elevation projected onto the image plane.

4.2 Sensor Alignment

The next step in the TC calibration is the alignment of the lidar and digital camera principal rays. The lidar is rotated about the x-axis (N) and the camera is rotated about the y-axis (E). The TC sensors can be aligned by rotation about both the x and y axes of the body frame. Alignment of the lidar about the x-axis is necessary because the true camera center found in Sec. 4.1 is not in the center of the digital image, but rather where the principal ray intersects the normalized image plane. Alignment about the body frame x-axis involves finding the lidar azimuth that aligns with the principal ray of the camera as seen in Fig. 4.5. Alignment about the y-axis is required due to the physical offset in the sensors that causes parallax, which affects the projection of a lidar point onto the image plane. Aligning the camera and lidar about the y-axis is shown in Fig. 4.6. Parallax can be compensated by aligning the lidar and the camera principal rays and then using equations described later in this section. Aligning the lidar and camera results in both principal rays being parallel. If the two sensors were coboresighted, with the same COP and principal ray, parallax would not be a concern. The correction for parallax is considered later in this section.

The method for aligning the lidar and camera can be done using the following steps:

1. Create a target to use as an alignment tool and place it perpendicular to the TC (Fig. 4.7).

2. Find the principal ray of the lidar and camera and align about the x-axis of the body frame.

3. Measure the physical offset between the sensors and the offset distance and distance to the alignment target to calculate the pixel offset between the camera and lidar.

4. Adjust the camera about the body frame y-axis to align its principal ray to be parallel with the lidar principal ray.

4.2.1 Alignment Process

The lidar principal ray extends from the lidar center, at an azimuth and elevation angle of zero, as seen by the red line in Fig. 4.5 and Fig. 4.6, respectively. The principal ray of the camera was found during the calibration in Sec. 4.1. Now, each sensor's principal rays must be aligned about the x and y axes. First, the TC is placed perpendicular to the calibration tool and at a distance greater than 25 m (Fig. 4.7). It should be mentioned that ideally, the sensors should be aligned at an infinite distance, but that cannot practically be done, so the further the alignment distance, the better the alignment. For the TC, it was decided that a distance greater than 25 m is sufficient.

Following step two above, the principal ray of the lidar and camera must be aligned about the x-axis of the body frame. To do this, the camera principal point found in Fig. 4.9, is marked on a digital image from the camera with a green crosshair (see Fig. 4.4) and the corresponding location is physically marked on the target with retro-reflective tape (reflects 850 nm EM waves) as illustrated in Fig. 4.8.

The retro-reflective tape creates a high intensity return when a lidar shot hits the tape. Because the camera principal point is offset from the center of the image, the column of lidar measurements that hits the retro-reflective target is selected to be the zero azimuth value, not necessarily the 1024^{th} lidar column. Because the lidar captures data over 360°, the sensor does not have to be rotated to align a column with the camera principal ray. Instead, the lidar azimuth that aligns with the retro-reflective target is considered the zero azimuth. This is illustrated in Fig. 4.9 and a real lidar scan taken from the TC showing the high intensity return is shown in Fig. 4.10.

After identifying the column of lidar measurements that aligns with the camera principal ray about the x-axis of the body frame, this column of lidar measurements is set to have an azimuth value of zero and will be called the zero azimuth in this document. The point at the center of this column is the center of the lidar normalized coordinate system ($\tilde{\boldsymbol{l}}_n$), given by

$$\tilde{\boldsymbol{l}}_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \tan\theta \\ \frac{\tan\phi}{\cos\theta} \end{bmatrix}, \tag{4.8}$$

where θ is the azimuth, and ϕ is the elevation in the lidar sensor coordinate frame. Measurements that occur before the zero azimuth (lidar points that are spatially to the left) have a negative azimuth value, and those that occur after the zero azimuth (spatially to the right) have positive values. Since the lidar and camera are mechanically offset in the elevation direction, the centers of projection must be offset by the same distance at the target. The next step is to align the lidar and camera about the y-axis of the body frame.

Aligning the lidar and camera to be parallel about the y-axis of the body frame requires the ability to adjust the position of the camera in the payload structure. For this alignment, the camera is pivoted around one of the mounting screws to make adjustments and align the camera to be parallel with the lidar (Fig. 4.11).

The alignment is complete when the camera principal ray points below the lidar principal and is parallel to it, as seen in Fig. 4.11. The distance the camera must point below the lidar is the same as the distance between the sensors. The number of pixels equivalent to this distance is calculated (see Appendix A.2), and a point is marked on the camera image with a red-crosshair (see Fig. 4.4), placed at the correct pixel offset from the principal point ((c_x, c_y) in (4.6)). Then the camera is pivoted until the red crosshair lines up with the retro-reflective tape placed at the lidar principal point on the calibration target. At this point, the alignment has been done, and a mapping of the lidar points to image pixels must be completed.

4.2.2 Mapping

After the sensors have been aligned, the mapping between lidar shots and image pixels must be estimated. This mapping technique is discussed in a previous paper [16]. For mapping, the TC is placed on a tripod, perpendicular to a flat surface at a distance z_1 from the camera. All of the lidar shots in a swath must fall on the surface of the wall. Using retro-reflective tape, mark N individual lidar points (typically 40 or more distributed throughout the camera FOV) on the wall. After marking N lidar points, an image is captured with all of the retro-reflective markings visible in the image. Then each lidar point is mapped to an image pixel by comparing the normalized lidar coordinates determined by (4.8) and the row-column values of the image pixels. The row-column values are then modified using

$$\begin{aligned}c &= g_1 + g_2\tilde{x}_n + g_3\tilde{x}_n^2 + g_4\tilde{x}_n^3 + g_5\tilde{y}_n + g_6\tilde{x}_n\tilde{y}_n + g_7\tilde{y}_n^2 + g_8\tilde{x}_n\tilde{y}_n^2 + g_9\tilde{x}_n^5 + g_{10}\tilde{x}_n\tilde{y}_n^4 + g_{11}\tilde{x}_n^3\tilde{y}_n^2\\ r &= h_1 + h_2\tilde{y}_n + h_3\tilde{y}_n^2 + h_4\tilde{y}_n^3 + h_5\tilde{x}_n + h_6\tilde{y}_n\tilde{x}_n + h_7\tilde{x}_n^2 + h_8\tilde{y}_n\tilde{x}_n^2 + h_9\tilde{y}_n^5 + h_{10}\tilde{y}_n\tilde{x}_n^4 + h_{11}\tilde{y}_n^3\tilde{x}_n^2,\end{aligned} \tag{4.9}$$

where $(c, r) = (\text{column}, \text{row})$ are the pixel coordinates and (x_n, y_n) are the lidar normalized coordinates. A maximum likelihood cost is used to fit the points, $\{(c, r)_i\} \leftrightarrow \{(x_n, y_n)_i\}$. An example of the mapping done for the TC is shown in Fig. 4.12, where the circles represent pixel locations obtained from the image and lidar point cloud, and the 'x' markers are estimated points based on the mapping algorithm.

As mentioned earlier, the effects of parallax must be compensated for by shifting the normalized lidar points in the in-track direction, due to the physical offset of the sensors. The compensation for parallax is only a function of z, the distance to an object in the z-direction of the body frame. Developing an equation requires the following variable definitions, z_1 is the distance from the TC to the calibration plane during mapping, d_0 is the physical offset in meters between the lidar and camera in the y direction, and z is the distance (meters) in the z-direction (body frame) for an individual lidar shot. Figure 4.13 shows an illustration of the parallax variables with reference to the TC.

Compensation shifts the normalized lidar coordinates in the in-track direction (Δy_n) based on the equation

$$\Delta y_n = \left(y_n + \frac{d_0}{z_1}\right) - \left(y_n + \frac{d_0}{z}\right), \tag{4.10}$$

which is simplified to be

$$\Delta y_n = d_0(\frac{1}{z_1} - \frac{1}{z}). \tag{4.11}$$

The parallax compensation in (4.11) shows that if an object is at the same range as the mapping distance z_1, then there is no shift in y_n ($\Delta y_n = 0$). Otherwise, the lidar point is shifted to account for a difference in view caused by parallax.

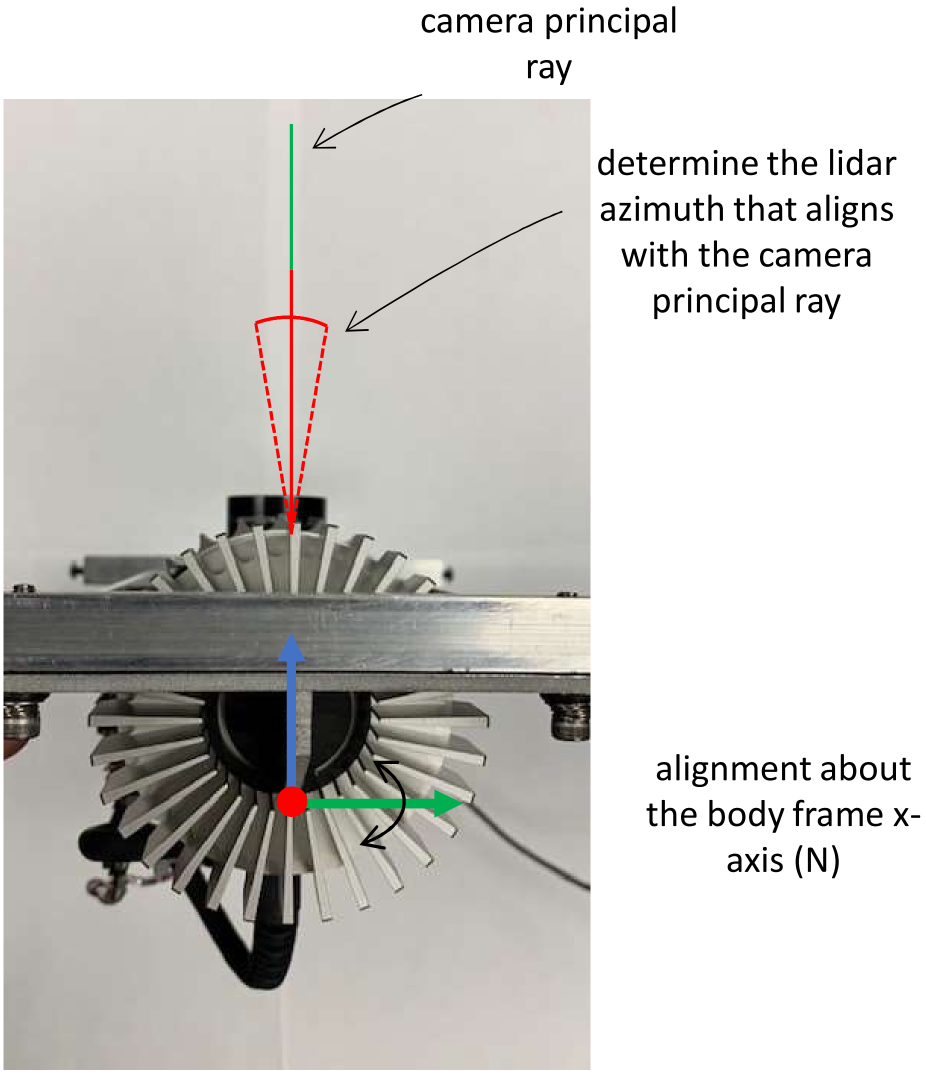

camera principal ray
determine the lidar azimuth that aligns with the camera principal ray
alignment about the body frame x-axis (N)

Fig. 4.5: Top-down view of the TC that shows how a lidar azimuth is selected to align with the principal ray of the camera about the x-axis of the body frame.

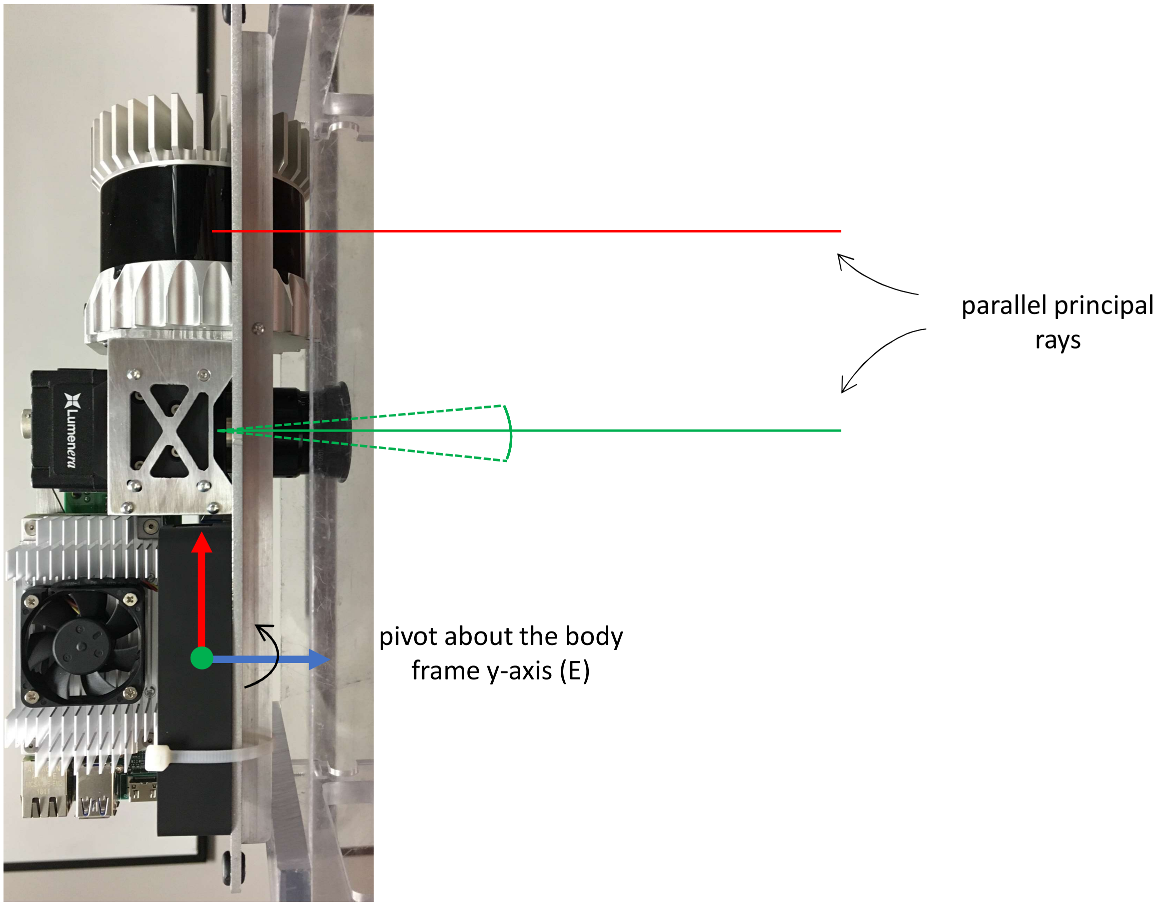


Fig. 4.6: Alignment of the camera and lidar principal rays about the body frame y-axis.

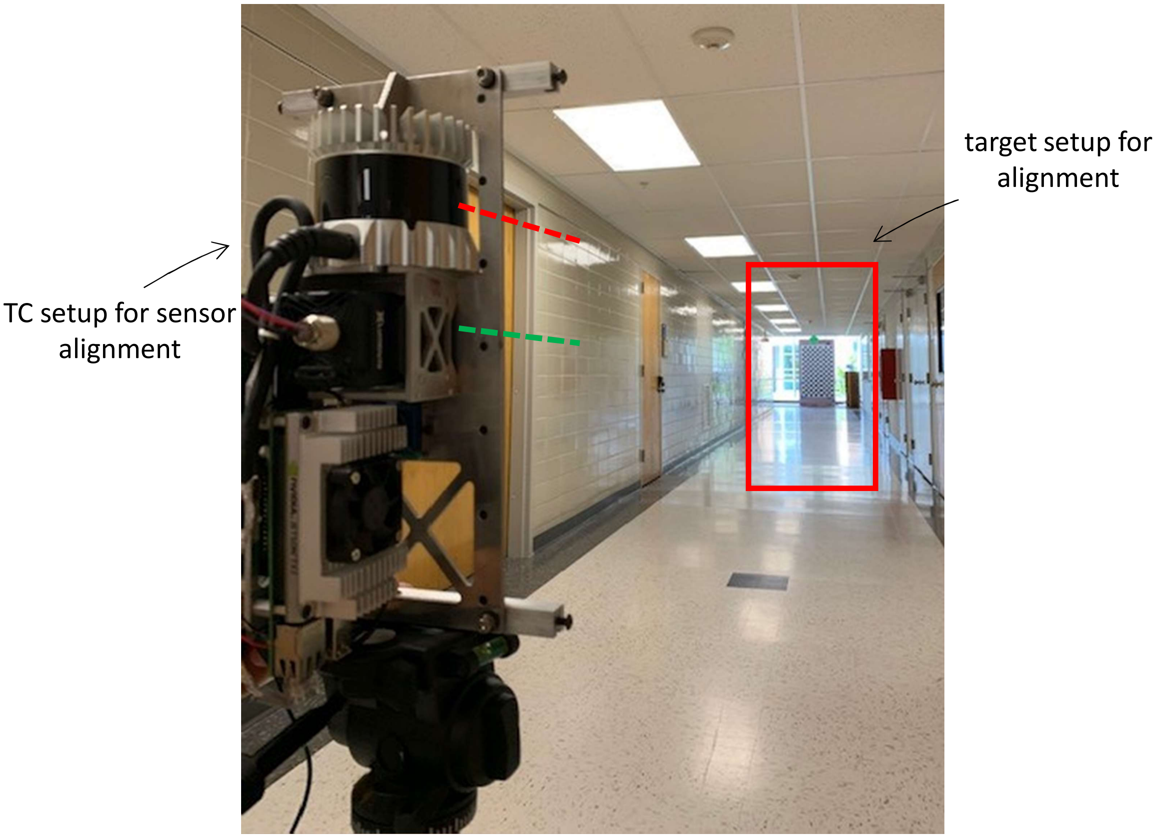


Fig. 4.7: TC setup for alignment on a tripod, perpendicular to the calibration tool.

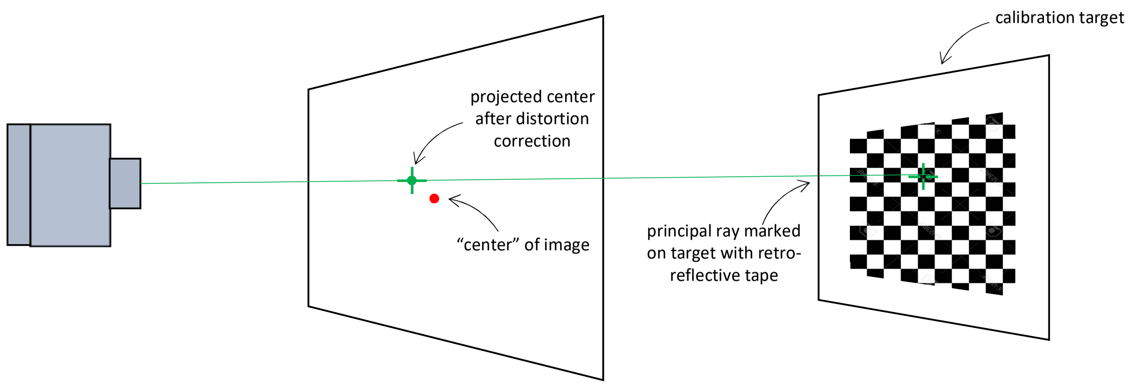

projected center after distortion correction
"center" of image
principal ray marked on target with retro-reflective tape
calibration target

Fig. 4.8: The camera principal ray, marked on the image plane with a green crosshair and marked on the calibration target with retro-reflective tape.

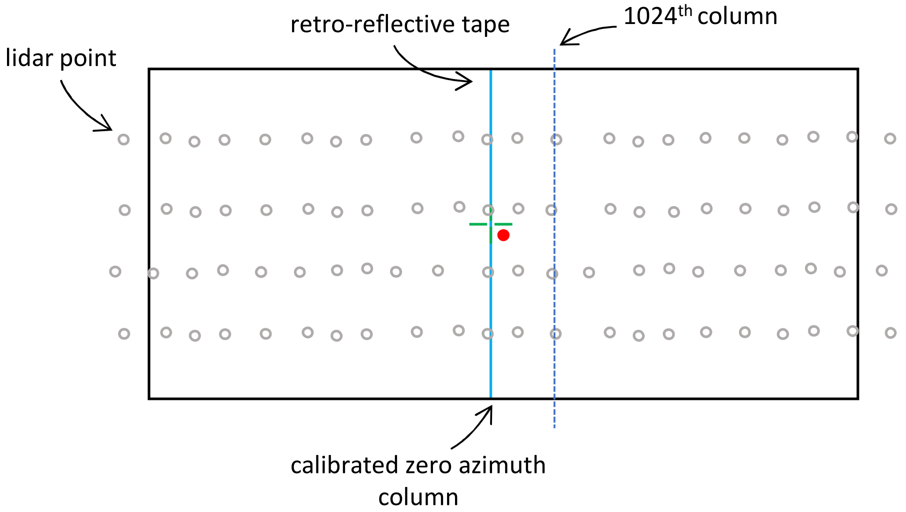


Fig. 4.9: An illustration of the calibration target with retro-reflective tape placed at the camera's principal point. This also shows a lidar scan and the column that falls along the retro-reflective tape, which is the column that is aligned with the camera principal ray and becomes the zero azimuth column.

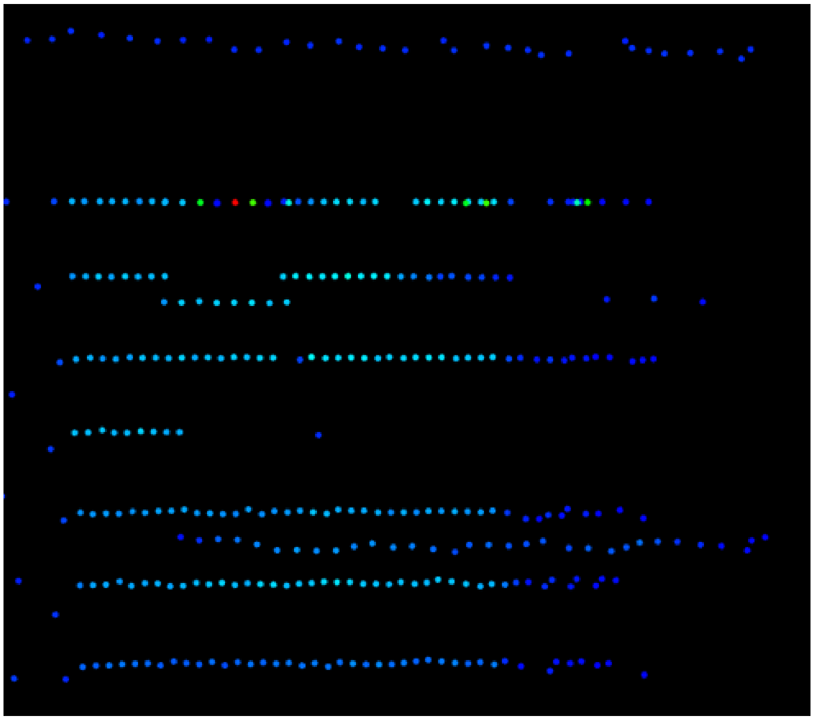


Fig. 4.10: A lidar scan from the TC (colored by intensity, with red being the highest intensity lidar shot return and blue being the least) showing the point cloud of the calibration target. There is a piece of retro-reflective tape on the target. One high intensity return can be seen as a red dot. The column of lidar measurements associated with the red dot becomes the TC calibrated zero azimuth column.

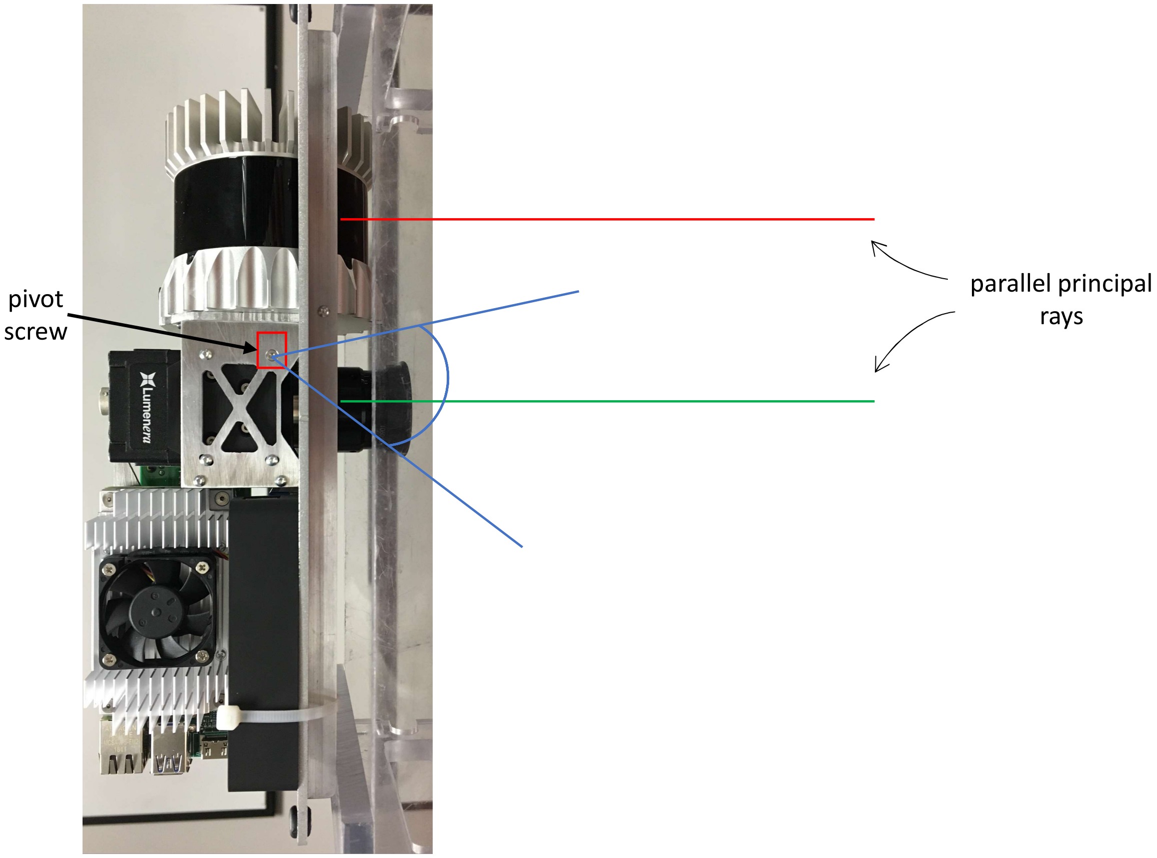
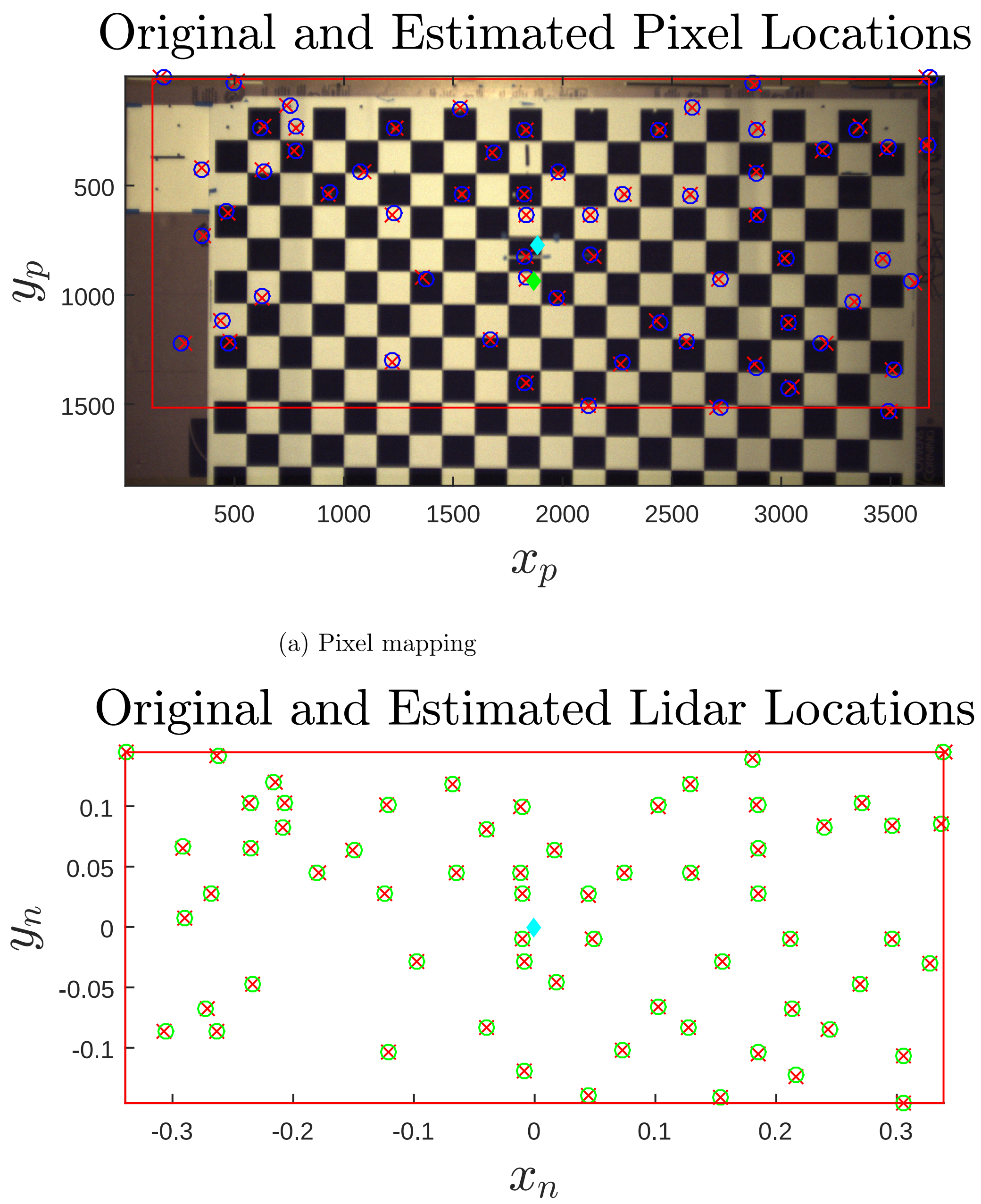


Fig. 4.11: The pivot screw (red box) and the angle the camera can be rotated about the pivot screw. The lidar principal ray is shown as the red line and the camera principal ray is shown as the green line. After alignment, these two lines are parallel to each other.



(a) Pixel mapping

(b) Lidar mapping

Fig. 4.12: Point mapping, where the circles represent points obtained from the digital image and lidar point cloud, and the "x" markers are the estimated point locations given the mapping. The blue diamond in (a) represents the image center while the green diamond is the principal ray found during calibration. (b) Normalized lidar points and the estimated values from mapping. The blue diamond represents the lidar principal ray.

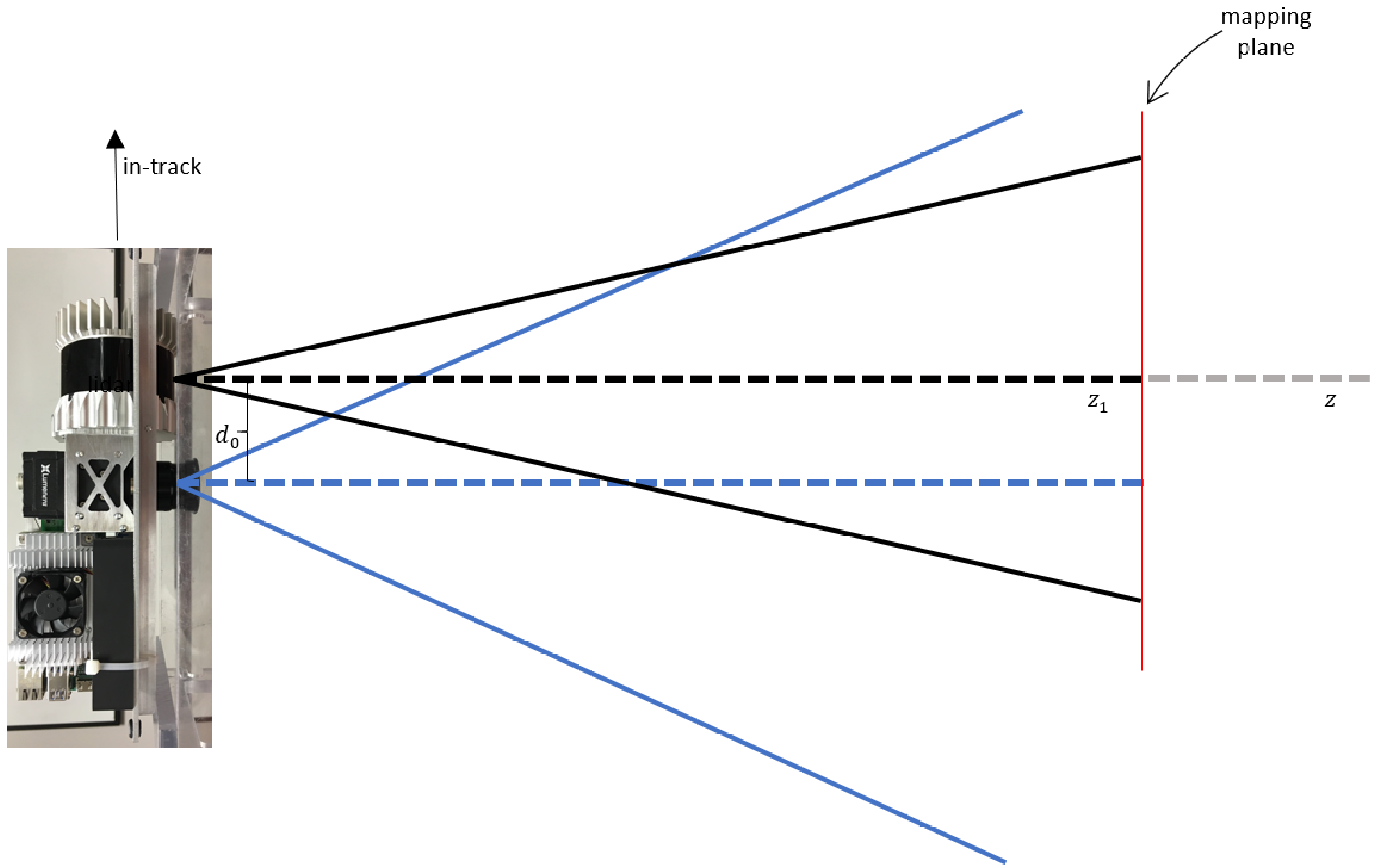


Fig. 4.13: Diagram of the TC parallax variables.

CHAPTER 5
CORRECTION FOR MOTION

5.1 Motivation for Correction

The lidar sensor has a rotation rate of 10 Hz (100 ms per rotation) and only collects range data for a portion of a scan. If the scan has a HFOV of 40° then the TC scan time (total amount of time that lidar range data is being collected during one lidar rotation) is approximately 11 ms. Typically, the image exposure lasts between 0.5 and 1 ms, so there is a significant difference in the duration of the TC scan time and the image exposure time. This means that there are always lidar shots captured before and after the image exposure time. When the TC is calibrated in a static environment, a lidar scan maps directly to the image pixels as described in Sec. 4.2.2, but when the TC is in motion, the lidar shots captured before the image exposure do not project to the same locations as they would if the environment was static. This same concept applies to the lidar shots taken after the image is exposed. So while the TC is in motion, the difference in the TC scan time and image exposure causes lidar points to be shifted at different locations throughout the scan, not just in a stationary environment as assumed during calibration. An illustration of how the lidar shots move during a scan is shown in Fig. 5.1.

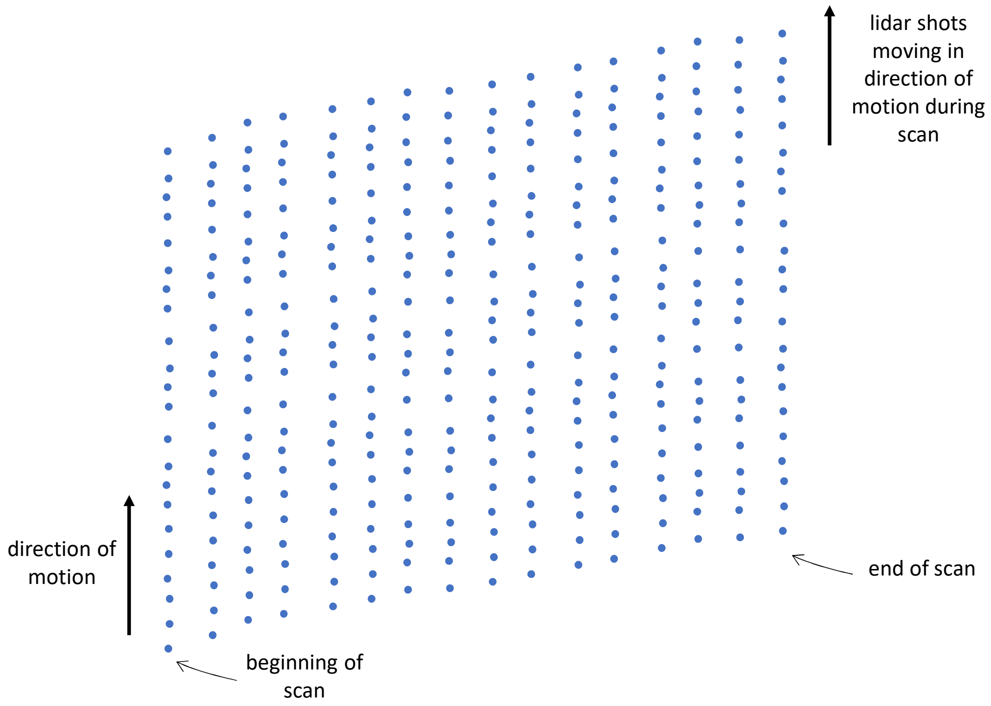


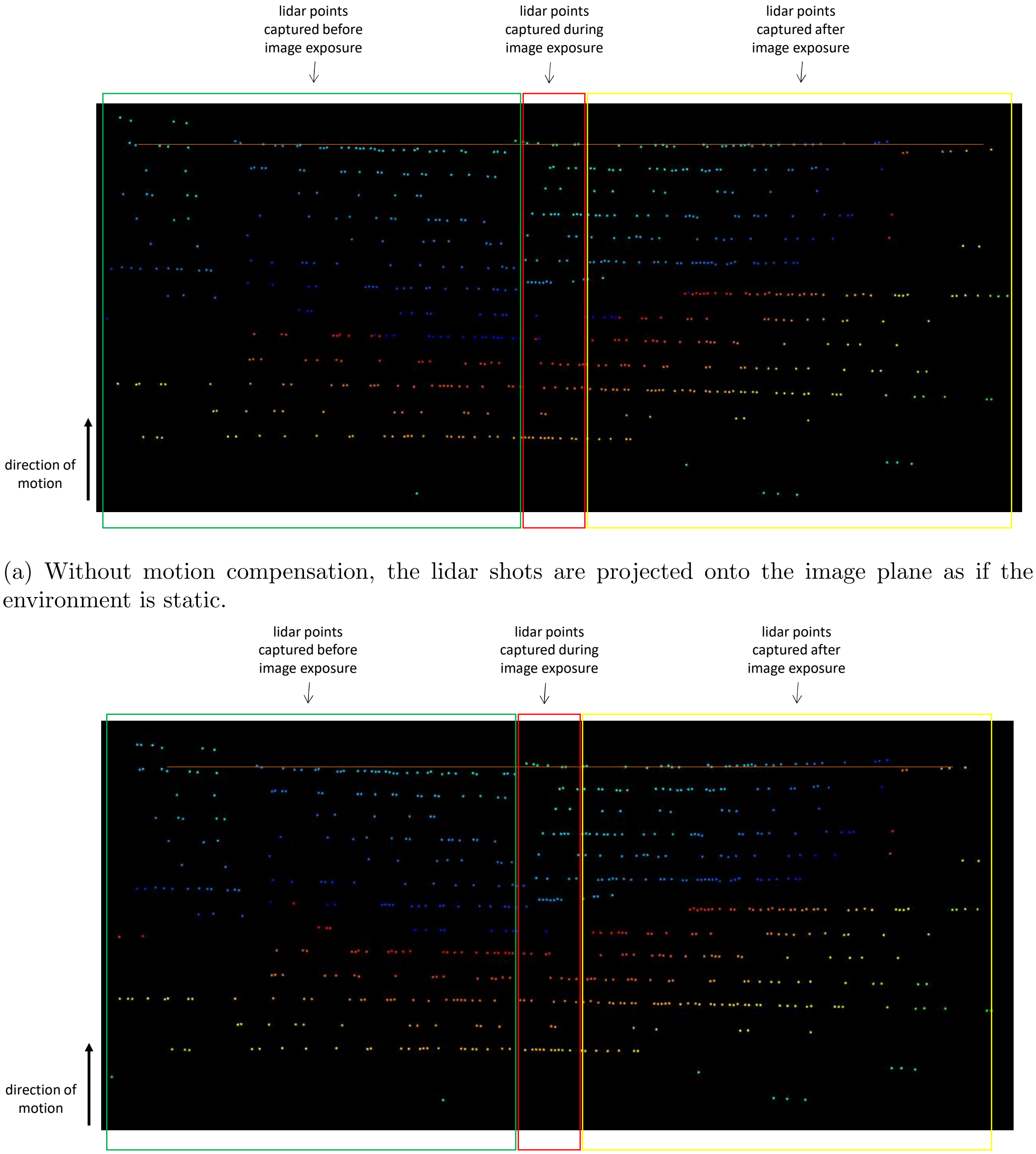
Fig. 5.1: Illustration of simulated data showing the motion of a lidar scan during flight.

When an image is captured during flight, some lidar shots are taken shortly before/after image exposure, as depicted in Fig. 5.2. For example, an sUAS moving at 15 m/s, the distance traveled on the ground in one TC swath is approximately 16 cm. If the image was taken immediately at the center of the scan, there is an 8 cm difference between where a lidar shot taken at the beginning (and end) of the scan should be. If the sUAS is flying at an altitude of 70m, shifting the normalized lidar point for motion results in a shift of 6 pixels in the image. For slow-moving applications, motion compensation is not necessary compared to fast-moving applications (mounted in a fixed-wing) where motion compensation increases the accuracy of a texel image.

It should be mentioned that if the lidar could capture an entire swath at the same time (for example, using flash lidar technology), motion compensation algorithms would not be needed. Due to the current state of flash lidar technology, there are no low-cost sensors that provide the range, frame rate, or resolution required by the TC.

By capturing inertial measurements throughout the TC swath, changes in position can be used to estimate the true location of the lidar (x, y, z) points, which is different from the static calibration due to the sUAS's motion. The inertial measurements containing position and attitude information can be interpolated for each lidar column time and the image capture time. Because lidar points are projected onto the image plane, each lidar shot must be related to the time of the image, and the position and attitude changes also must be accounted for to project a lidar shot at time t_l onto the image at time t_c.

To show the shift in lidar points due to motion, a lidar scan taken from the TC in Fig. 5.2 shows the static calibration compared to the motion correction and how lidar shots taken before the image time shift. The orange line is a control line, perpendicular to the direction of motion, in the same location for each lidar scan. In the static calibration scan, most of the lidar shots in a single row align with the orange line. In the motion correction scan, the shots taken before the image time show a shift down in the scan while the shots taken after the image time show a shift up in the scan. This is an example of how the lidar shots move due to the motion of the sUAS.



(a) Without motion compensation, the lidar shots are projected onto the image plane as if the environment is static.

(b) The illustration shows the actual positions of the lidar shots relative to the image time due to motion.

Fig. 5.2: The lidar scan in (a) shows no motion vs. (b) motion compensation. Lidar shots captured before the image time are shown in the green box, lidar shots captured after the image time are shown in the yellow box, and the red box containing of lidar shots taken during image exposure.

It can be seen from Fig. 5.2 that motion correction may increase the accuracy of texel images captured from the TC. The ideas and algorithms explaining the correction are explained in the following sections.

5.2 Motion Correction Algorithm

Errors in mapping lidar shots to image pixels during motion occur because the static mapping does not move lidar points to their position relative to the image acquisition time. To compensate for motion, a method was developed to relate each lidar shot's position at the time the image was acquired. Each lidar column time is associated with a position in the world coordinate system and orientation. Because there are few inertial measurements compared to the number of lidar columns, linear interpolation is done to determine the position and orientation of the TC for all lidar columns in the scan. Then the equations to project a lidar point position at any time t_i to the position at the time of the camera image acquisition t_c are derived. To do this, some concepts and notations must be introduced.

First, let us define the variables used to describe the interpolation of position, orientation, and lidar point projection. All variables used in this section are seen in Table 5.1.

Variable Name	**Description**
t_{n-1},t_n,t_{n+1},...	Time of an INS pose measurement.
t_l	Time of a lidar column.
t_c	Time the camera is exposed.
$\tilde{\boldsymbol{l}}^L(t_l)$	Lidar (x, y, z) measurement in the lidar sensor frame at the time of a lidar column.
$\boldsymbol{l}^L(t_c)$	Line of sight vector of the lidar at the time the camera was exposed (t_c).
$\boldsymbol{p}^{ENU}(t_i)$	Position of the body at the time t_i (can be t_l, t_c, etc).
$\boldsymbol{q}^{ENU}_{ins}(t_i)$	Quaternion of the body frame represented in the inertial frame (ENU). at a time t_i
$\boldsymbol{p}^{ENU}_f$	Position of a feature in the world coordinate system (ENU).
$R^{ENU}_b(t_l)$	Rotation matrix, at a specific lidar column time (t_l), to rotate a coordinate from the body frame of the INS to ENU.
$R^b_L(t_i)$	Rotation matrix used to rotate a coordinate from the lidar sensor frame to the body frame of the INS. This rotation matrix is a constant because of the placement of the senors on the mechanical structure.
$\boldsymbol{d}^b$	Displacement from the body of the INS sensor to the center of the lidar sensor.

Table 5.1: Variables used during motion compensation and their variable descriptions

Figure 5.3 shows the data acquisition events of the TC while the sUAS is in multiple poses during one swath. Although distances are exaggerated, this illustrates the variables introduced above.

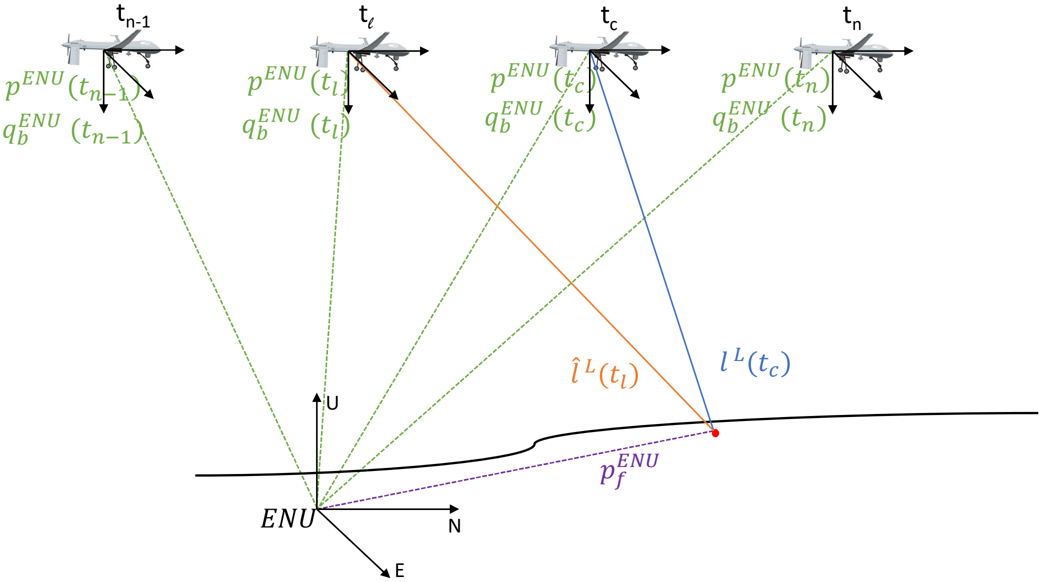


Fig. 5.3: The data acquisition events during a TC swath and the relative positions of the TC at each time.

Also, because there is a physical offset between the INS and the lidar sensor, a lever arm must be added to the position vector $p^{ENU}(t_i)$. The lever arm distance $\boldsymbol{d}^b$ has nonzero x and z components, but zero y component because of the placement of the sensors described in Sec. 3.2.2. The diagram with the addition of a lever arm between sensors is in Fig. 5.4.

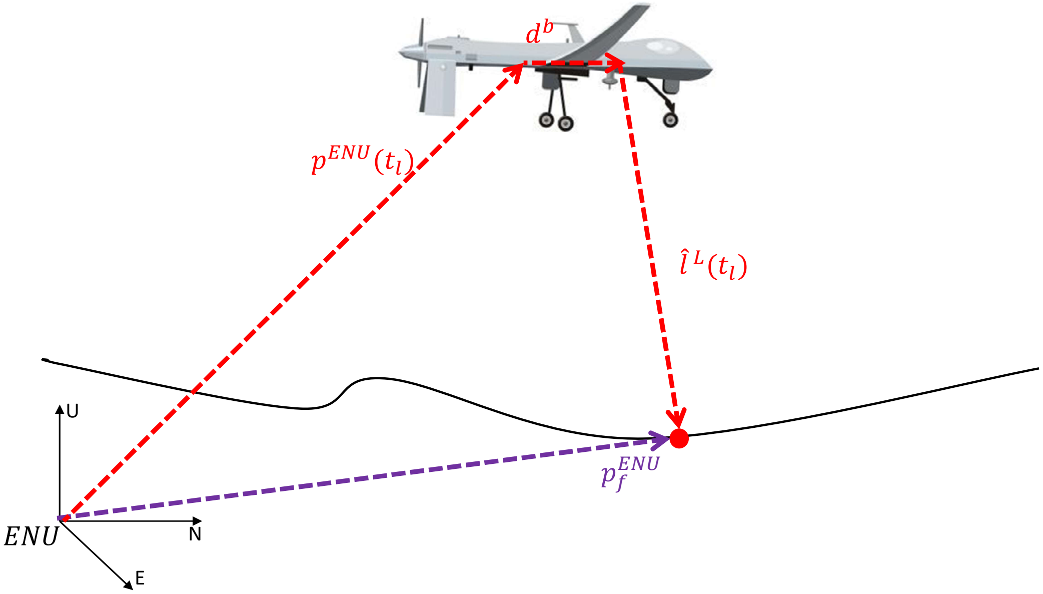


Fig. 5.4: The physical offset between the INS and lidar introduce a lever arm that must be taken into account.

The value of the lever arm for the TC is given by

$$\boldsymbol{d}^b = \begin{bmatrix} 0.111 & 0 & -0.004 \end{bmatrix}^T [\text{m}]. \tag{5.1}$$

5.2.1 Linear Interpolation of Pose

The process for linear interpolation of pose is explained in the following steps:

1. Select the first two INS pose measurements and interpolate the pose of every lidar column that occurs between the times of the two INS poses. For example, in Fig. 5.3, the two INS measurements that will be used are at times t_{n-1} and t_n.

2. Step through all of the INS packets until every lidar column has a position and orientation associated with it.

3. If there are lidar columns that occur before the first pose measurement or after the last pose, extrapolate the pose of each lidar column using the closest interpolated

segment.

Starting with the linear interpolation of position for all lidar columns, determine $\boldsymbol{p}^{ENU}(t_i)$ at any intermediate time t_i. Linear interpolation is a straightforward concept and is given by

$$\boldsymbol{p}^{ENU}(t_i) = \boldsymbol{p}^{ENU}(t_{n-1}) + \frac{\boldsymbol{p}^{ENU}(t_n) - \boldsymbol{p}^{ENU}(t_{n-1})}{t_n - t_{n-1}}(t_i - t_{n-1}). \tag{5.2}$$

Now to determine the interpolated orientation, θ, the total rotation angle from the one orientation to the the next orientation given by (5.4). This derivation uses normalized quaternions, and care must be taken to ensure proper interpolation. As a note, the symbol $\otimes$ represents quaternion multiplication, but for simplification purposes, it may not be included in all equations. It should be clear based on context when one quaternion multiplies another that quaternion multiplication must be applied.

Interpolating the orientation is described by

$$\boldsymbol{q}_{ins}^{ENU}(t_i) = \delta\boldsymbol{q}(t_i) \otimes \boldsymbol{q}_{ins}^{ENU}(t_{n-1}), \tag{5.3}$$

which describes a small rotation from $\boldsymbol{q}_{ins}^{ENU}(t_{n-1})$ to get the orientation at any intermediate time $\boldsymbol{q}_{ins}^{ENU}(t_i)$, where

$$\delta\boldsymbol{q}(t_i) = \begin{bmatrix} 1 \\ \frac{\boldsymbol{\theta}(t_i)}{2} \end{bmatrix}. \tag{5.4}$$

Now (5.4) must be solved for the variable $\boldsymbol{\theta}(t_i)$. To do this, an equation relating the initial and final orientations, t_{n-1} and t_n respectively, is defined by

$$\boldsymbol{q}_{ins}^{ENU}(t_n) = \delta\boldsymbol{q}(t_n) \otimes \boldsymbol{q}_{ins}^{ENU}(t_{n-1}), \tag{5.5}$$

where

$$\delta \boldsymbol{q}(t_n) = \begin{bmatrix} 1 \\ \frac{\Delta\boldsymbol{\theta}}{2} \end{bmatrix}. \tag{5.6}$$

Solve for $\delta\boldsymbol{q}(t_n)$ to find $\Delta\boldsymbol{\theta}$ and multiply by the conjugate of $\boldsymbol{q}_{ins}^{ENU}(t_{n-1})$ (denoted with an asterisk) on both sides of the equation to obtain

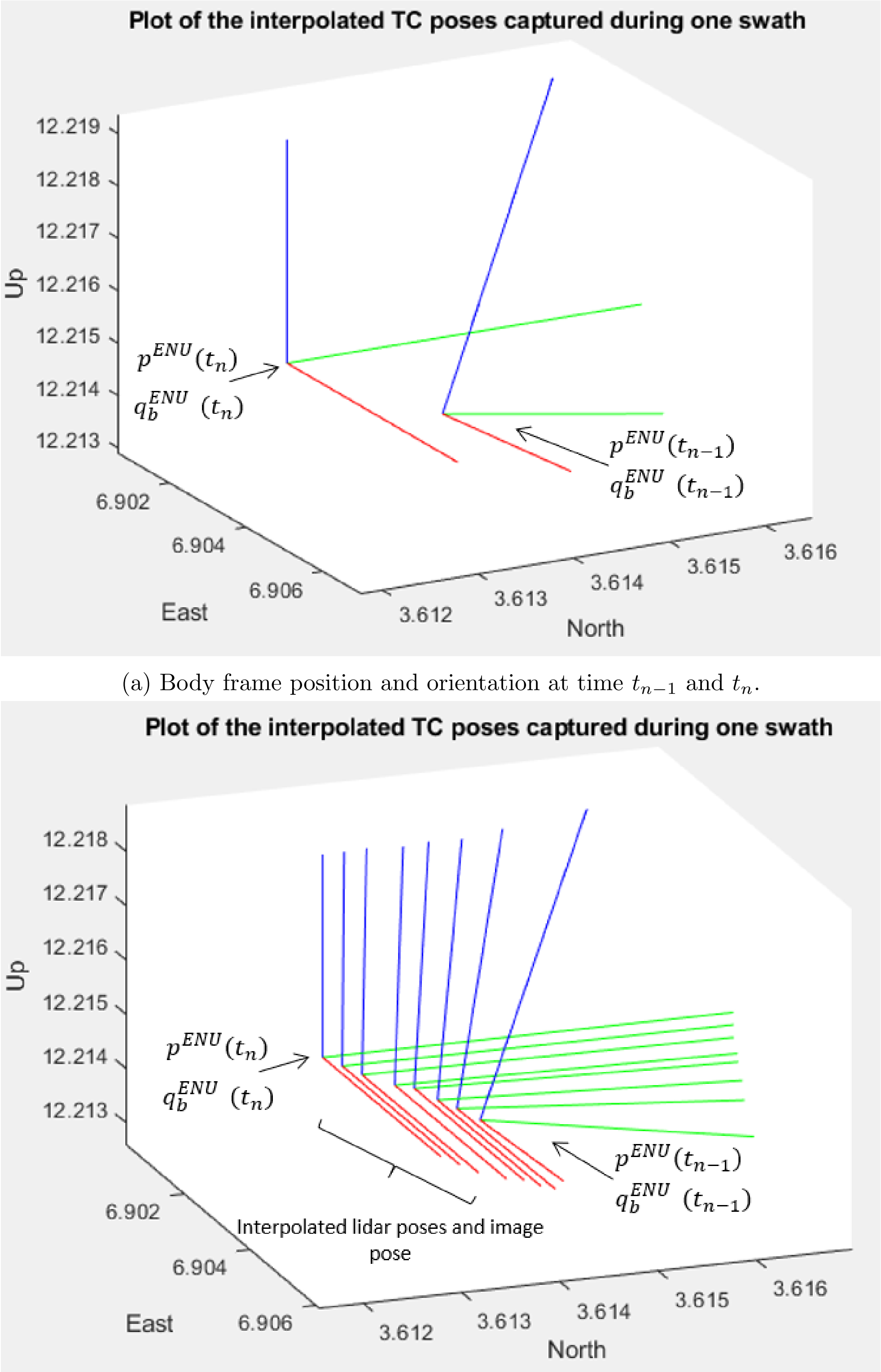
$$\boldsymbol{q}_{ins}^{ENU}(t_n) \otimes (\boldsymbol{q}_{ins}^{ENU}(t_{n-1}))^{*} = \delta\boldsymbol{q}(t_n) = \begin{bmatrix} 1 \\ \frac{\Delta\boldsymbol{\theta}}{2} \end{bmatrix}. \tag{5.7}$$

At this point, $\Delta\boldsymbol{\theta}$ has been identified as the rotation from $\boldsymbol{q}_{ins}^{ENU}(t_{n-1})$ to $\boldsymbol{q}_{ins}^{ENU}(t_n)$ and can be used in the linear interpolation given by

$$\boldsymbol{\theta}(t_i) = \frac{\Delta\boldsymbol{\theta}}{t_n - t_{n-1}}(t_i - t_{n-1}) \tag{5.8}$$

to determine $\boldsymbol{\theta}(t_i)$ used in (5.4).

Using the equations described in this section, the interpolation of position and pose of each lidar column time can be computed. The interpolation was done on real TC data captured during an experimental flight and plotted to show the interpolation of the pose in Fig. 5.5(a) and Fig. 5.5(b). The 3D plot in Fig. 5.5(a) shows two recorded positions and poses at times t_n and t_{n-1} taken by the INS during flight. After completing the interpolation of position and pose between the two times, the 3D plot in Fig. 5.5(b) shows the set of positions and poses interpolated between the times t_n and t_{n-1}. These equations complete the interpolation of the pose, and the next step is to use these positions and orientations to project the lidar point to the image plane correctly.



(a) Body frame position and orientation at time t_{n-1} and t_n.

(b) Interpolation of the lidar poses from the body frame position and orientation at time t_{n-1} and t_n in 5.5(a.)

Fig. 5.5: MATLAB plots of experimental TC pose data being interpolated using the methods described in this section.

5.2.2 Point Projection Using Motion Correction

Using Fig. 5.3 and 5.4 it is easy to see that the position of the feature $\boldsymbol{p}_f^{ENU}$ can be solved for in terms of measurements made at the time of a lidar column (t_l) using

$$\boldsymbol{p}_f^{ENU} = \boldsymbol{p}^{ENU}(t_l) + R_b^{ENU}(t_l)R_L^b\tilde{\boldsymbol{l}}^L(t_l) + R_b^{ENU}(t_l)\boldsymbol{d}^b, \tag{5.9}$$

where

$\boldsymbol{p}_f^{ENU}$ is the xyz location of a feature in the inertial frame,

$\boldsymbol{p}^{ENU}(t_l)$ is the xyz location of the TC in the inertial frame at the time of an individual lidar shot (t_l),

$R_b^{ENU}(t_l)$ is the rotation matrix from the body frame to the inertial frame at t_l,

R_L^b is the rotation matrix from lidar frame to the body frame,

$\tilde{\boldsymbol{l}}^L(t_l)$ is a lidar xyz measurement at time t_l, expressed in the lidar frame at the time t_l, and

$\boldsymbol{d}^b$ is the lever arm from the body frame to the lidar frame.

It is essential to note an assumption made throughout this document. It is assumed that the position of a feature does not change during a scan. This is a sound assumption when mapping objects such as buildings, rocks, and other stationary terrain, but fails when there are moving objects such as pedestrians.

Since each lidar point must be projected onto the image plane at the time the image was taken, using Fig. 5.3 and 5.4, an equation for the position of the same feature $\boldsymbol{p}_f^{ENU}$ in (5.9) can be found at the camera time t_c, resulting in

$$\boldsymbol{p}_f^{ENU} = \boldsymbol{p}^{ENU}(t_c) + R_b^{ENU}(t_c)R_L^b\boldsymbol{l}^L(t_c) + R_b^{ENU}(t_c)\boldsymbol{d}^b. \tag{5.10}$$

Since we have two equations for the feature location, we can set them equal to each other and solve for the line of sight vector $\boldsymbol{l}^L(t_c)$. Setting (5.9) and (5.10) equal to each other results in

$$\boldsymbol{p}^{ENU}(t_l)+R_b^{ENU}(t_l)R_L^b\tilde{\boldsymbol{l}}^L(t_l)+R_b^{ENU}(t_l)\boldsymbol{d}^b=\boldsymbol{p}^{ENU}(t_c)+R_b^{ENU}(t_c)R_L^b(t_c)\boldsymbol{l}^L(t_c)+R_b^{ENU}(t_c)\boldsymbol{d}^b. \tag{5.11}$$

Solving (5.11) for $\boldsymbol{l}^L(t_c)$ gives

$$\boldsymbol{l}^L(t_c)=R_b^L(t_c)R_{ENU}^b(t_c)\left(\boldsymbol{p}^{ENU}(t_l)-\boldsymbol{p}^{ENU}(t_c)+R_b^{ENU}(t_l)R_L^b\tilde{\boldsymbol{l}}^L(t_l)+R_b^{ENU}(t_l)\boldsymbol{d}^b-R_b^{ENU}(t_c)\boldsymbol{d}^b\right) \tag{5.12}$$

The normalized lidar coordinates are found by projecting the 3D points from the lidar LOS vector in (5.12), onto the lidar image plane. The normalized lidar coordinates are defined as $\boldsymbol{u}(t_c)$, and are given by

$$\boldsymbol{u}(t_c)=\begin{bmatrix}\frac{l_x^L(t_c)}{l_z^L(t_c)}\\ \frac{l_y^L(t_c)}{l_z^L(t_c)}\\ 1\end{bmatrix}=\begin{bmatrix}x_n(t_c)\\ y_n(t_c)\\ 1\end{bmatrix}. \tag{5.13}$$

From this point, the normalized coordinates are modified for parallax and mapping as described in Sec. 4.1 and Sec. 4.2.2.

For the TC software, the process above is modified to use quaternions rather than rotation matrices (quaternion conventions described in [32]). The reasons for this are because the INS provides pose information in quaternions and there are computational benefits for using quaternions, including reduced number of multiplications when compared to rotation matrices. The process produces the same results but modifies the notation of the equations because of the nature of quaternion operations. To do the above computations using quaternions rather than rotation matrices, let Q be a rotation matrix, let $\boldsymbol{q}$ be a quaternion, let $\boldsymbol{w}$ be a 3x1 vector in R^3, and $\boldsymbol{v}\in R^3$ but is represented by a pure quaternion with zero as the scalar to be $\begin{bmatrix}0 & \boldsymbol{v}^T\end{bmatrix}^T$. The following equations continue to use vectors defined in R^3 for quaternion multiplication and should be treated as pure quaternions. Using the properties

$$\begin{bmatrix} 0 \\ Q\boldsymbol{w} \end{bmatrix} = \boldsymbol{q}\boldsymbol{v}\boldsymbol{q}^* \tag{5.14}$$

and

$$\begin{bmatrix} 0 \\ Q^T\boldsymbol{w} \end{bmatrix} = \boldsymbol{q}^*\boldsymbol{v}\boldsymbol{q} \tag{5.15}$$

defined in [32], and applying them to (5.12) gives

$$\begin{aligned} \boldsymbol{l}^L(t_c) = \boldsymbol{q}^L_{ENU}(t_c) \Big(\boldsymbol{p}^{ENU}(t_l) - \boldsymbol{p}^{ENU}(t_c) + \boldsymbol{q}^{ENU}_L(t_l) \left(\tilde{\boldsymbol{l}}^L(t_l)\right) \boldsymbol{q}^{*ENU}_{L}(t_l) + \\ \boldsymbol{q}^{ENU}_L(t_l)\boldsymbol{d}^b\boldsymbol{q}^{*ENU}_{L}(t_l) - \boldsymbol{q}^{ENU}_L(t_c)\boldsymbol{d}^b\boldsymbol{q}^{*ENU}_{L}(t_c)\Big) \boldsymbol{q}^{*L}_{ENU}(t_c) \end{aligned} \tag{5.16}$$

All of the steps discussed up to this point can be summarized to complete the motion correction algorithm as seen in Algorithm 1.

Algorithm 1: Motion Correction Algorithm

Repeat the algorithm between each pose measurement $(\boldsymbol{p}(t_n), \boldsymbol{p}(t_{n-1}), \boldsymbol{p}(t_{n-2}), ...$ and $\boldsymbol{q}_L^{ENU}(t_n), \boldsymbol{q}_L^{ENU}(t_{n-1}), \boldsymbol{q}_L^{ENU}(t_{n-2}), ...)$ captured in a TC swath.

1. Linear interpolation of the TC position at the time of a lidar shot taken between INS measurements

$$\boldsymbol{p}^{ENU}(t_l) = \boldsymbol{p}^{ENU}(t_{n-1}) + \frac{\boldsymbol{p}^{ENU}(t_n) - \boldsymbol{p}^{ENU}(t_{n-1})}{t_n - t_{n-1}}(t_l - t_{n-1}) \tag{5.2}$$

2. Linear interpolation of the TC orientation at the time of a lidar shot between INS measurements

$$\boldsymbol{q}_L^{ENU}(t_l) = \delta\boldsymbol{q}(t_l) \otimes \boldsymbol{q}_L^{ENU}(t_{n-1}) \tag{5.3}$$

3. Use the interpolated pose for each lidar shot to determine the LOS of the lidar at the time of camera exposure

$$\begin{aligned}\boldsymbol{l}^L(t_c) = \boldsymbol{q}_{ENU}^L(t_c)\Big(&\boldsymbol{p}^{ENU}(t_l) - \boldsymbol{p}^{ENU}(t_c) + \boldsymbol{q}_L^{ENU}(t_l)\left(\tilde{\boldsymbol{l}}^L(t_l)\right)\boldsymbol{q}^{*}{}_L^{ENU}(t_l) + \\ &\boldsymbol{q}_L^{ENU}(t_l)\boldsymbol{d}^b\boldsymbol{q}^{*}{}_L^{ENU}(t_l) - \boldsymbol{q}_L^{ENU}(t_c)\boldsymbol{d}^b\boldsymbol{q}^{*}{}_L^{ENU}(t_c)\Big)\boldsymbol{q}^{*}{}_{ENU}^L(t_c)\end{aligned} \tag{5.16}$$

4. Find the projections onto the normalized lidar image plane

$$\boldsymbol{u}(t_c) = \begin{bmatrix} \frac{l_x^L(t_c)}{l_z^L(t_c)} \\ \frac{l_y^L(t_c)}{l_z^L(t_c)} \\ 1 \end{bmatrix} \tag{5.13}$$

5.3 Experimental Results

Algorithm 1 was tested by acquiring more than 5000 swaths of data while the TC was mounted on a rotorcraft. Post-processing was done on the data with and without the correction algorithm to compare the results of the correction. The processing and texel image generation was done using C++. During operation, the rotorcraft flew over various buildings on the Utah State University campus at speeds up to 13 m/s. In theory, the motion correction algorithm would show the most significant improvements when the sUAS is moving the fastest. There are multiple sections in a flight plan where a rotorcraft has a low velocity, sometimes even being stationary before starting another part of the flight path.

To compare the static calibration method to the motion correction method, every image captured by the TC was marked at the pixel locations where a lidar point is projected. The static calibration method projects the lidar point on the image as a red dot, and the motion correction method projects a blue dot. Many figures in this section have these dots to allow a comparison of the original lidar shot projection versus the corrected lidar projection onto the image plane.

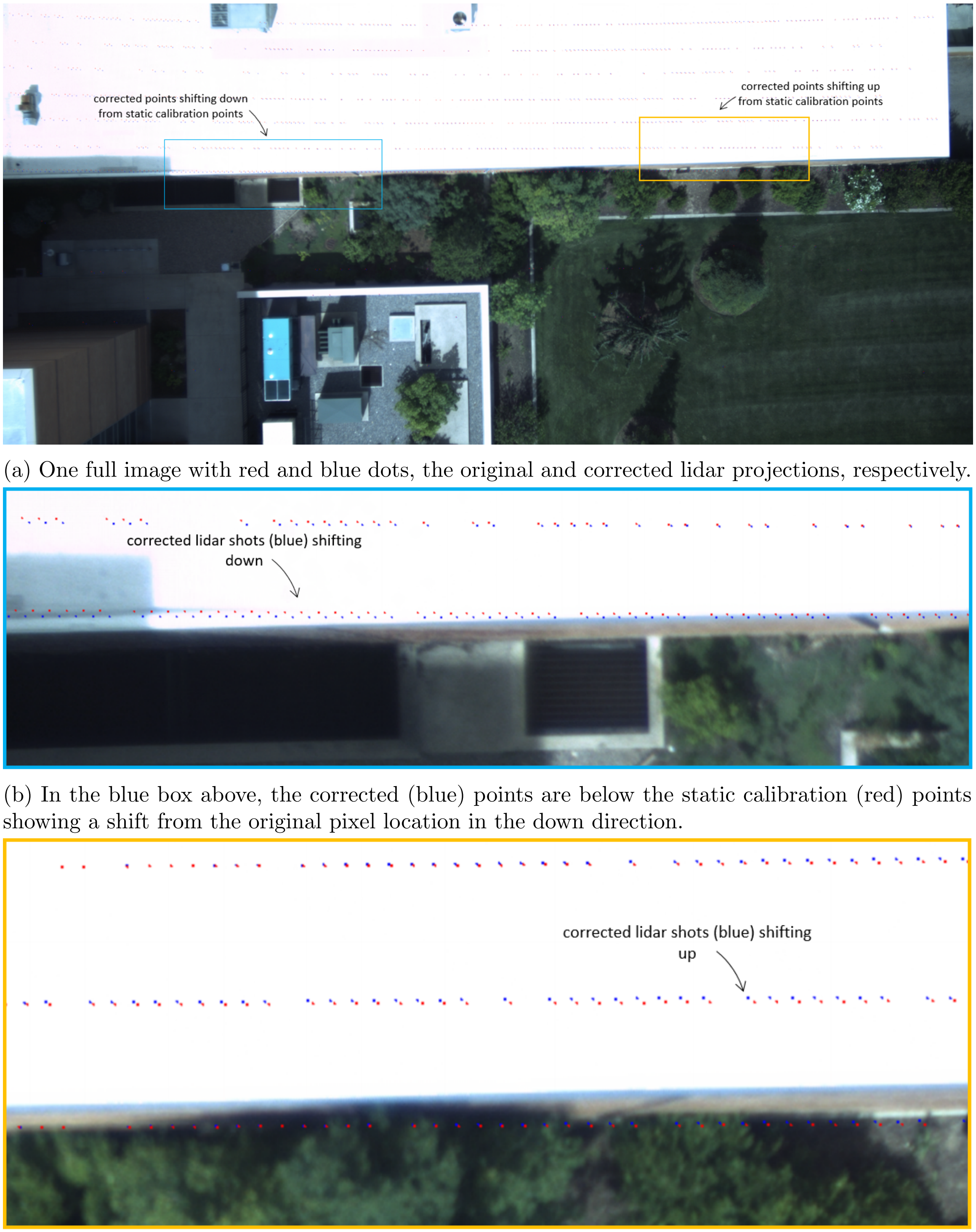
The correction doesn't dramatically change every point but there are specific locations in a texel image that are most impacted that can be studied in more detail. For instance, any lidar shot that occurs before or after the time the image is captured is corrected in the normalized image plane as a function of the time difference between the lidar shot and the image capture time. So, as this difference increases, the correction becomes greater. This indicates that the most significant corrections occur at the edges of a scan when a lidar shot is captured at the maximum time difference from the image capture time in a swath. Also, the greatest translational correction occurs when the sUAS is at high velocity and the greatest rotational correction occurs during large attitude changes. Understanding these principles, swaths that match this criteria can be analyzed to see improvements.

One of the first indications that the algorithm is correct is if the lidar points shift based on the expectation in Fig. 5.2b, where lidar shots captured before the capture time appear shifted down in the image and points captured after the image time appear shifted up in the image. Figure 5.6 shows one image captured by the TC while the sUAS is in motion and has red and blue points on the image that represent where a lidar shot is projected on the image plane for the static calibration and motion correction, respectively. Figure 5.6(a) is the full digital image and includes blue and orange rectangles that correspond to Fig. 5.6(b) and 5.6(c) respectively. These figures show how the corrected lidar shots shift in the image plane.

Specific scenarios that benefit from the correction are the edges of buildings and objects captured at the edges of a texel image. An example of this is shown in Fig. 5.7, which captures the corner of a building where lidar shots land on the roof and the sidewalk next to the building. This case is common when imaging suburban environments with lots of buildings. The motion correction improvements can be seen by comparing the static calibration of the texel image in Fig. 5.7(b) to the corrected texel image in Fig. 5.7(c). The comparison shows that the correction algorithm aligns the edges of the rooftop with the lidar shots landing on the roof and there is no stretching of the rooftop down the wall.

Improvements can be seen on another building corner shown in Fig. 5.9(a). This shows the static calibration in Fig. 5.9(b) compared to the corrected texel image in Fig. 5.9(c). The scenario is exactly the same as described previously in Fig. 5.7 where the the TC is capturing data on a rooftop near edges and corners. It is clear that these situations are some of the most common areas to see improvements in texel images with the correction algorithm applied.

In many cases, the TC experiences attitude disturbances as well as changes in rotation when making turns. An example of various texel images taken in succession are shown in Fig. 5.11. From Fig. 5.11, the texel images show a 90 degree rotation of the TC. This maneuver is common for rotorcraft to align the sUAS along the next flight path. The corrected images can be seen in Fig. 5.12 where the corrected lidar shots are shifted from the original lidar projections shown as red dots.

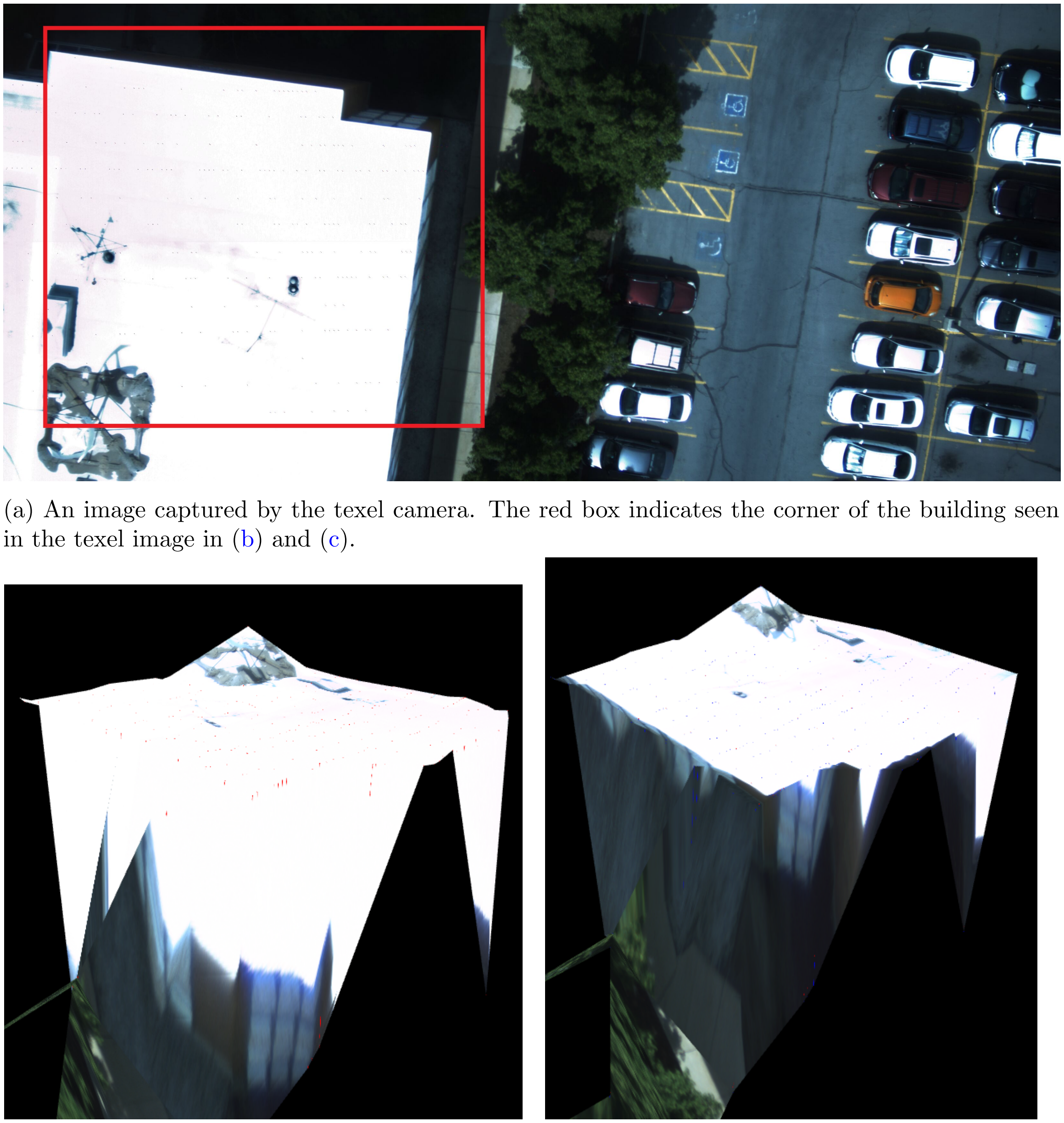

corrected points shifting down
from static calibration points
corrected points shifting up
from static calibration points
corrected lidar shots (blue) shifting
down
corrected lidar shots (blue) shifting
up

(a) One full image with red and blue dots, the original and corrected lidar projections, respectively.

(b) In the blue box above, the corrected (blue) points are below the static calibration (red) points showing a shift from the original pixel location in the down direction.

(c) In the orange box above, the corrected (blue) points move up from the static calibration (red) points in the image with the direction of motion.

Fig. 5.6: Verification of experimental results by inspecting how the corrected (blue) lidar points move in the image.

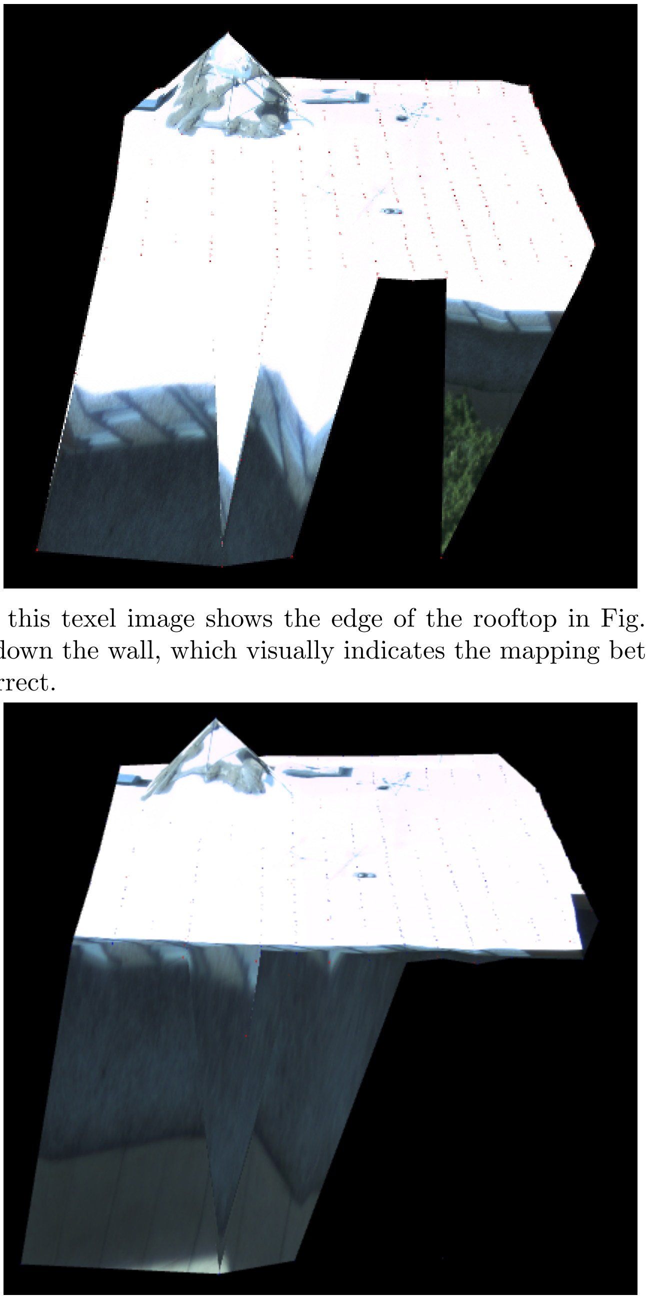


(a) An image captured by the texel camera. The red box indicates the corner of the building seen in the texel image in (b) and (c).

(b) Before correction, this texel image shows the corner of a rooftop with the white roof texture pulled down along the wall of the building. It is obvious that this creates a poor visual texture.

(c) After correction, this texel image shows how the corrected normalized lidar coordinates shift to correct the mapping between lidar shots and image pixels.

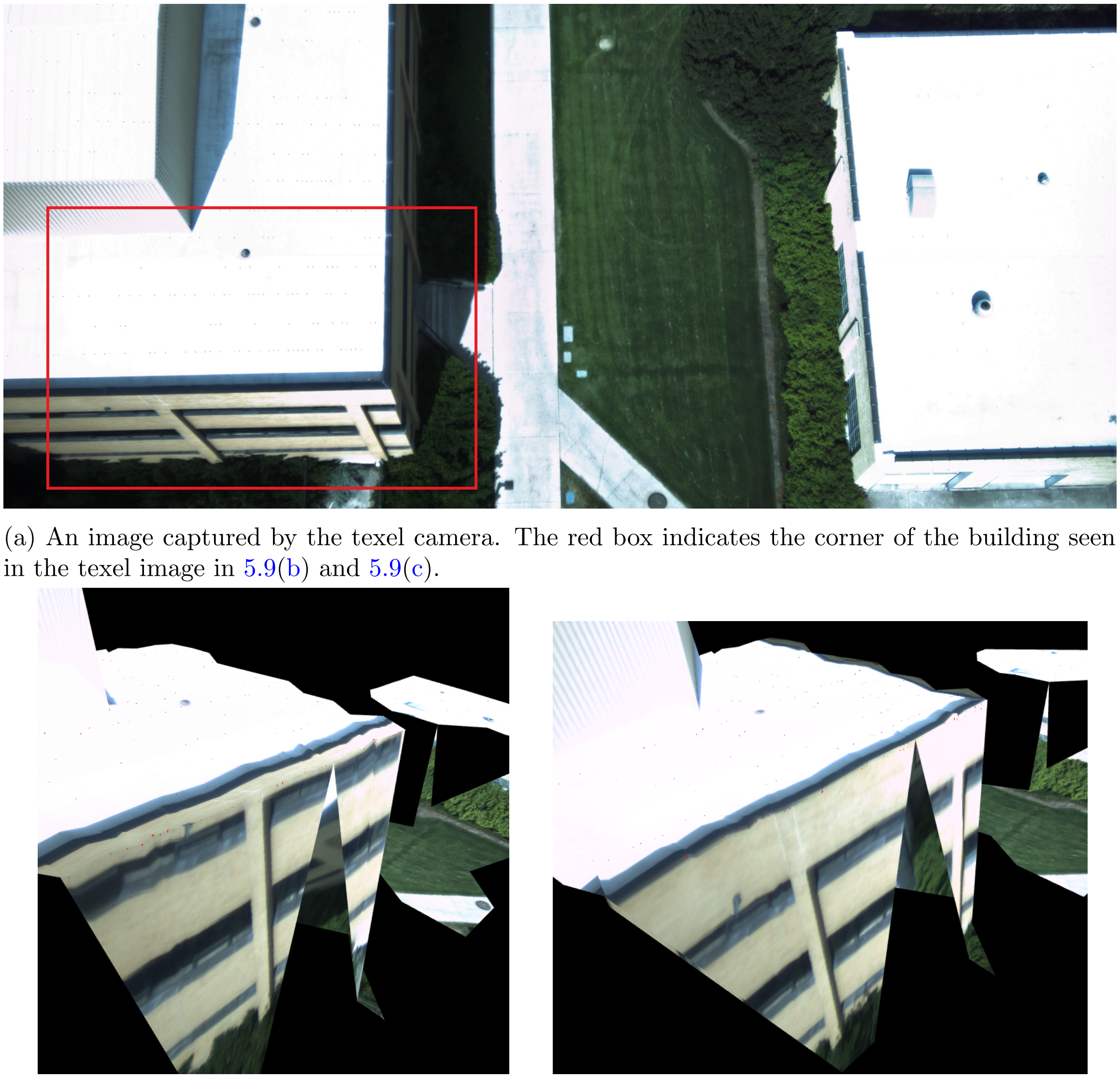
Fig. 5.7: Figure 5.7(a) shows an image taken with the TC with a red box indicating the subsection of the image located in 5.7(b) and 5.7(c).



(a) Before correction, this texel image shows the edge of the rooftop in Fig. 5.7(a). The texture on the roof is sliding down the wall, which visually indicates the mapping between lidar points and image pixels is not correct.

(b) After correction, this texel image shows how the corrected normalized lidar coordinates project on the normalized image plane.

Fig. 5.8: This is another view of a texel image similar to the position in Fig. 5.7(a). In Fig. 5.8(a), the original texel image shows the texture of the corner of the building sliding down on the wall in the texel image and 5.8(b) is the corrected texel image with a defined corner.

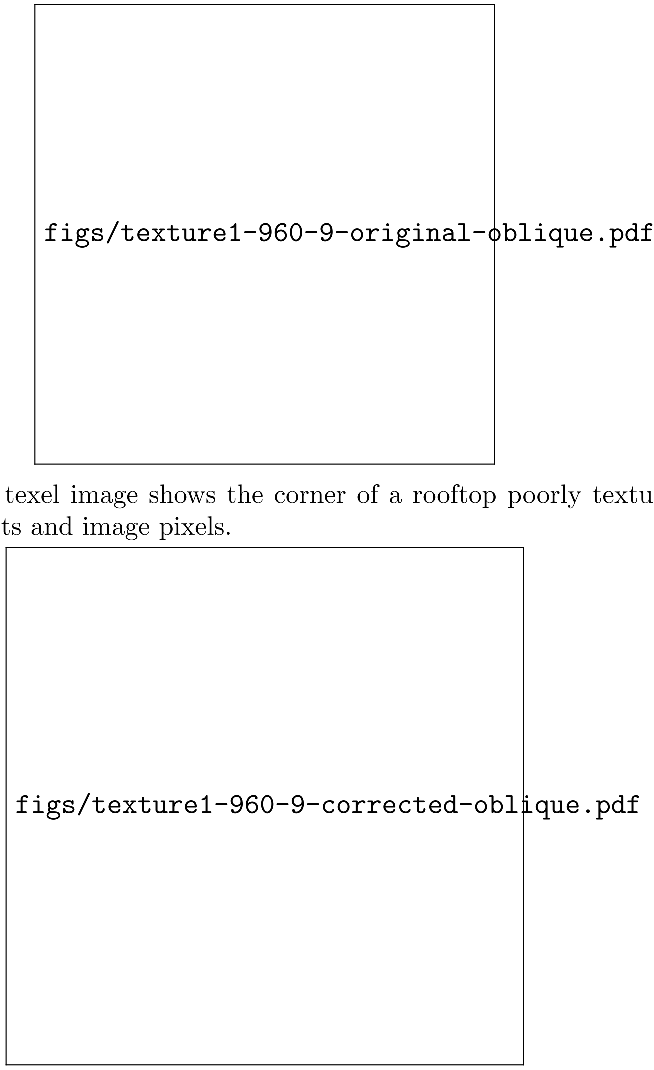


(a) An image captured by the texel camera. The red box indicates the corner of the building seen in the texel image in 5.9(b) and 5.9(c).

(b) Before correction, this texel image shows the corner of a rooftop with the white roof texture pulled down along the wall of the building.

(c) After correction, this texel image shows the how the corrected normalized lidar coordinates shift to correct the mapping between lidar shots and image pixels.

Fig. 5.9: Figure 5.9(a) shows an image taken with the TC with a red box indicating the subsection of the image located in 5.9(b) and 5.9(c).


figs/texture1-960-9-original-oblique.pdf
figs/texture1-960-9-corrected-oblique.pdf

(a) Before correction, this texel image shows the corner of a rooftop poorly textured due to poor mapping between lidar shots and image pixels.

(b) This texel image shows the corrected normalized lidar coordinates projected onto the normalized image plane to provide an improved mapping between lidar shots and image pixels.

Fig. 5.10: This is another view of the texel image from Fig. 5.9(a). In 5.10(a), the original texel image shows the texture of the corner of the building sliding down on the wall in the texel image and 5.10(b) is the corrected texel image with a defined corner.

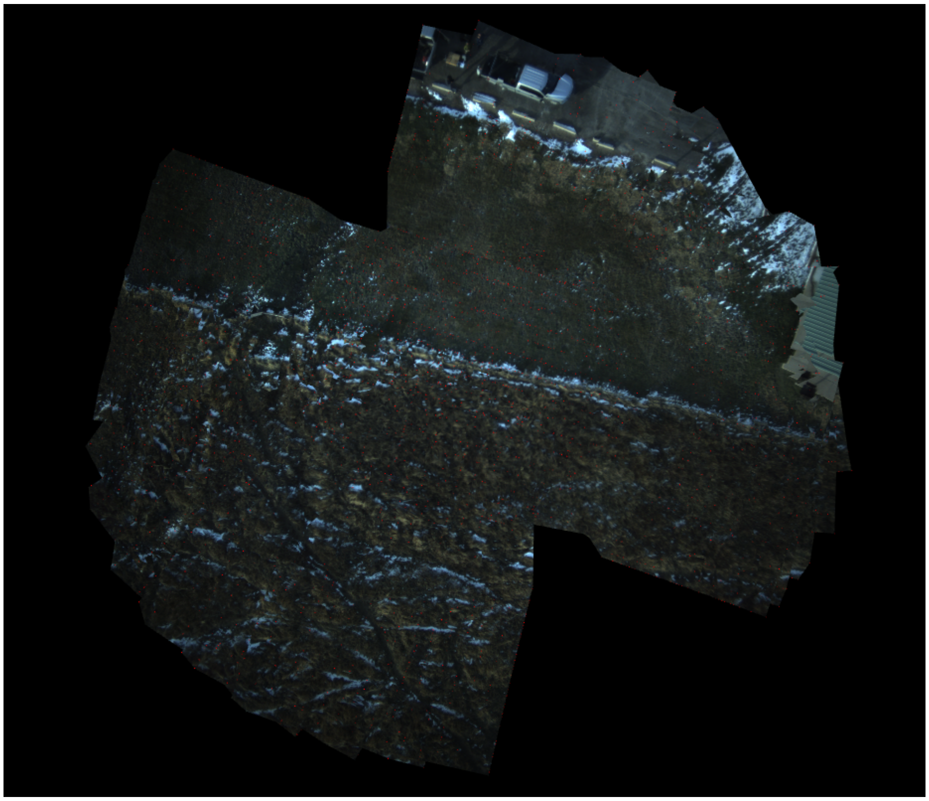
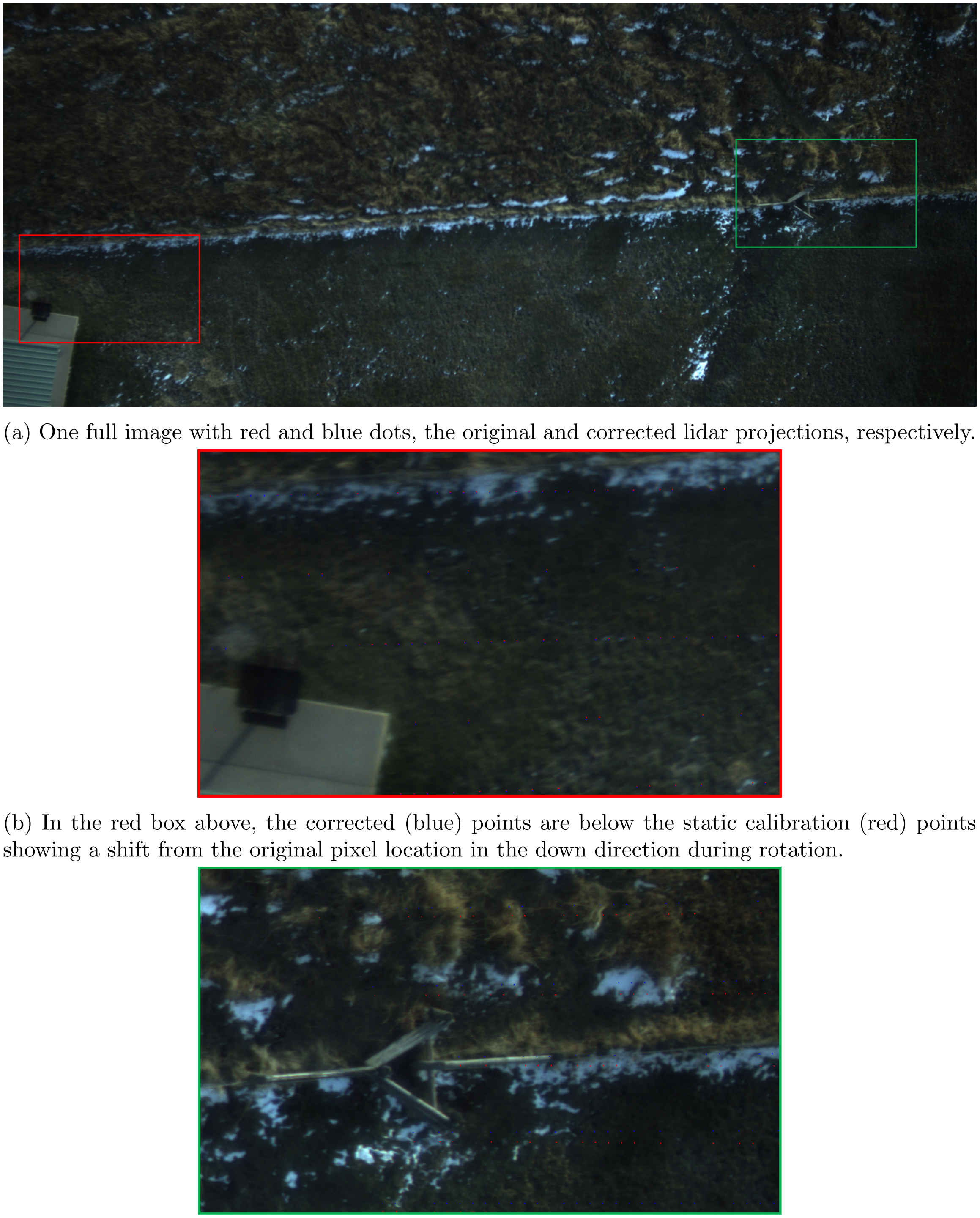


Fig. 5.11: Multiple texel images of a field, fence line, and a parking lot during rotation of the TC during flight. The images show a 90 degree turn from the original pose.



(a) One full image with red and blue dots, the original and corrected lidar projections, respectively.

(b) In the red box above, the corrected (blue) points are below the static calibration (red) points showing a shift from the original pixel location in the down direction during rotation.

(c) In the green box above, the corrected (blue) points move up from the static calibration (red) points in the image with the direction of rotation.

Fig. 5.12: Verification of experimental results by inspecting how the corrected (blue) lidar points move in the image during rotation.

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

This thesis discusses the development of a TC, mounted in a sUAS for the acquisition and development of TDSMs. While the idea of a sensor to combine lidar and digital imagery is not entirely new, the methods for developing the TC are unique. Along with the TC's mechanical design, each sensor must be chosen to meet specifications while considering factors such as cost, size, weight, and power. The TC mechanical structure provides a rigid frame to reduce vibration effects and is designed to adjust the sensors for accurate calibration.

The algorithm and methods of calibration discussed in this thesis are unique compared to other methods by including corrections for image distortion (unlike many autocalibration methods), developing a compensation for parallax, and the mapping techniques previously developed for the TC [16]. Although it is not an automatic calibration, taking a more involved approach described in Chapter 4 can accurately calibrate a lidar sensor and camera for a robust 3D data set. The calibration of lidar and image sensors for the creation of 3D data sets will continue to be an important process in remote sensing. Even in the case of a flash lidar, the need for calibration does not become obsolete and must always be accounted for when developing sensors that combine range and texture data.

The algorithm developed for corrections provides a straightforward approach to add motion compensation and improves accuracy in scenarios with high velocity and large range differences on the edge of a TC swath. The motion compensation algorithm shows the potential to increase accuracy in applications with 13 m/s or higher velocities.

One drawback to the motion compensation algorithm presented in this thesis is that as lidar technology improves to include flash lidar with high resolution, frame rate, and accuracy, the algorithm essentially becomes obsolete. Currently, it seems that most lidar manufacturers with rotational lidar sensors are making advances to existing rotational sensors that will allow this algorithm to be used in future TC versions built with rotational lidar.

The development of the TC shows the increase in Technology Readiness Level (TRL) from a level 3 to a level 5, from the original handheld TC described in [28], to a completed system with successful flight operations.

6.2 Future Work

Future implementations of the TC should include a lidar sensor with a larger range and multiple returns to provide more feature information and texture terrain beneath the foliage. Because the current lidar sensor only has one return, it limits the TC from utilizing the full benefits of multiple return lidar. Multiple returns would create textured digital elevation models (TDEMs), rather than surface models.

Further research could also be done to improve the ability to align sensors at a distance of 50 or more meters. The further the alignment distance, the better the sensors are aligned, which increases the TC's overall accuracy. The reason this is difficult is because of the sensitivity of the camera to adjustments at a long-range. Simple adjustments cause substantial differences in the calibration at a longer range.

REFERENCES

[1] M. Simard, N. Pinto, J. B. Fisher, and A. Baccini, “Mapping forest canopy height globally with spaceborne lidar,” *Journal of Geophysical Research: Biogeosciences*, vol. 116, no. 4, dec 2011.

[2] J. Gené-Mola, E. Gregorio, J. Guevara, F. Auat, R. Sanz-Cortiella, A. Escolà, J. Llorens, J. R. Morros, J. Ruiz-Hidalgo, V. Vilaplana, and J. R. Rosell-Polo, “Fruit detection in an apple orchard using a mobile terrestrial laser scanner,” *Biosystems Engineering*, vol. 187, pp. 171–184, 2019.

[3] R. Zhang, B. Yang, W. Xiao, F. Liang, Y. Liu, and Z. Wang, “Automatic Extraction of High-Voltage Power Transmission Objects from UAV Lidar Point Clouds,” *Remote Sensing*, vol. 11, no. 22, p. 2600, nov 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/22/2600

[4] “Center for Advanced Imaging LADAR at Utah State University.” [Online]. Available: https://engineering.usu.edu/ece/cail/

[5] “AggieAir Group at Utah State University,” 2019. [Online]. Available: https://aggieair.usu.edu/

[6] M. Takahashi, K. Kobayashi, K. Watanabe, and T. Kinoshita, “Development of prediction based emergency obstacle avoidance module by using LIDAR for mobile robot,” *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems, SCIS 2014 and 15th International Symposium on Advanced Intelligent Systems, ISIS 2014*, pp. 561–564, 2014.

[7] H. Guan, W. Yan, Y. Yu, L. Zhong, and D. Li, “Robust Traffic-Sign Detection and Classification Using Mobile LiDAR Data With Digital Images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 5, pp. 1715–1724, may 2018.

[8] K. Chiang, G. Tsai, Y. Li, and N. El-Sheimy, “Development of LiDAR-Based UAV System for Environment Reconstruction,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1790–1794, oct 2017.

[9] C. Hug, A. Ullrich, and A. Grimm, “Litemapper-5600-a waveform-digitizing LiDAR terrain and vegetation mapping system,” *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. Part 8, p. W2, 2004.

[10] K. Lim, P. Treitz, M. Wulder, B. St-Onge, and M. Flood, “LiDAR remote sensing of forest structure,” *Progress in Physical Geography: Earth and Environment*, vol. 27, no. 1, pp. 88–106, 2003. [Online]. Available: https://doi.org/10.1191/0309133303pp360ra

[11] C. Kiemle, A. Schäfler, M. Wirth, A. Fix, and S. Rahm, “Detection and Analysis of Water Vapor Transport by Airborne Lidars,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 3, pp. 1189–1193, jun 2013.

[12] J. R. G. Braga, H. F. d. C. Velho, and E. H. Shiguemori, “Estimation of UAV position using LiDAR images for autonomous navigation over the ocean,” in *2015 9th International Conference on Sensing Technology (ICST)*. IEEE, dec 2015, pp. 811–816. [Online]. Available: http://ieeexplore.ieee.org/document/7438508/

[13] A. Mathews and J. Jensen, “Visualizing and Quantifying Vineyard Canopy LAI Using an Unmanned Aerial Vehicle (UAV) Collected High Density Structure from Motion Point Cloud,” *Remote Sensing*, vol. 5, no. 5, pp. 2164–2183, may 2013. [Online]. Available: http://www.mdpi.com/2072-4292/5/5/2164

[14] S. Schneider, M. Himmelsbach, T. Luettel, and H. J. Wuensche, “Fusing vision and LIDAR - Synchronization, correction and occlusion reasoning,” *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 388–393, 2010.

[15] S. Xie, D. Yang, K. Jiang, and Y. Zhong, “Pixels and 3-D Points Alignment Method for the Fusion of Camera and LiDAR Data,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 10, pp. 3661–3676, oct 2019.

[16] S. E. Budge, N. S. Badamikar, and X. Xie, “Automatic registration of fused lidar/digital imagery (texel images) for three-dimensional image creation,” *Optical Engineering*, vol. 54, no. 3, p. 031105, 2014.

[17] T. C. Bybee and S. E. Budge, “Method for 3-D Scene Reconstruction Using Fused LiDAR and Imagery from a Texel Camera,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 8879–8889, 2019.

[18] L. Wang and U. Neumann, “A robust approach for automatic registration of aerial images with untextured aerial LiDAR data,” *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2623–2630, 2010.

[19] J. S. Berrio, M. Shan, S. Worrall, J. Ward, and E. Nebot, “Semantic sensor fusion: from camera to sparse lidar information,” 2020. [Online]. Available: http://arxiv.org/abs/2003.01871

[20] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder (improves camera calibration),” *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2301–2306, 2004.

[21] G. Li, Y. Liu, L. Dong, X. Cai, and D. Zhou, “An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 3854–3859, 2007.

[22] G. Pandey, J. McBride, S. Savarese, and R. Eustice, *Extrinsic calibration of a 3D laser scanner and an omnidirectional camera*. IFAC, 2010, vol. 7, no. PART 1. [Online]. Available: http://dx.doi.org/10.3182/20100906-3-IT-2019.00059

[23] J. Castorena, U. S. Kamilov, and P. T. Boufounos, "Autocalibration of lidar and optical cameras via edge alignment," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2016-May, pp. 2862–2866, 2016.

[24] A. Kassir and T. Peynot, "Reliable automatic camera-laser calibration," *Proceedings of the 2010 Australasian Conference on Robotics and Automation, ACRA 2010*, 2010.

[25] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3936–3943, 2012.

[26] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3D lidar and camera by maximizing mutual information," *Proceedings of the National Conference on Artificial Intelligence*, vol. 3, pp. 2053–2059, 2012.

[27] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," *Proceedings - IEEE International Conference on Robotics and Automation*, no. c, pp. 3685–3691, 2013.

[28] B. M. Boldt, S. E. Budge, R. T. Pack, and P. D. Israelsen, "Handheld Carnera for Acquiring Irnages," pp. 0–4.

[29] M. Treiber, "Relationship between airborne lidar product specifications and end product requirements," no. May 2019, p. 3, 2019.

[30] J. Heikkila and O. Silvén, "Calibration procedure for short focal length off-the-shelf CCD cameras," *Proceedings - International Conference on Pattern Recognition*, vol. 1, pp. 166–170, 1996.

[31] Http://www.vision.caltech.edu/bouguetj/calib_doc/, "Camera Calibration Toolbox for MATLAB." [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib{_}doc/

[32] J. B. Kuipers, *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*, 1999, vol. 37, no. 01.

[33] Https://en.wikipedia.org/wiki/Angle_of_view, "Angle of View - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Angle{_}of{_}view

APPENDICES

APPENDIX A

A.1 Camera FOV Calculation

As mentioned in Chapter 3, the camera sensor FOV is computed based on the configured lidar horizontal and vertical FOV. This calculation relies on the characteristics of the camera lens focal length and camera sensor size. The equation describing a camera's field of view is [33]

$$\alpha = 2 * arctan\left(\frac{d}{2f}\right) \tag{A.1}$$

where α is the angle, d is the sensor size in the direction measured, and f is the effective focal length (16440 μm). The sensor size is dependent on the size of an individual sensor (3.1 μm) and the number of sensors N in the vertical or horizontal direction (4240 horizontal and 2832 vertical) as described by

$$d = 3.1\mu m * N \tag{A.2}$$

$$HFOV_{max} = 2 * arctan\left(\frac{3.1\mu m * 4240}{2 * 16440\mu m}\right) = 43.5788^\circ \tag{A.3}$$

$$VFOV_{max} = 2 * arctan\left(\frac{3.1\mu m * 2832}{2 * 16440\mu m}\right) = 29.8993^\circ \tag{A.4}$$

Using $HFOV_{max}$ and $VFOV_{max}$ and knowing the configured FOV for the lidar, the image size can be computed by substituting in the desired α and solving for the number of pixels N.

A.2 Distance to Pixels

The equations to determine the number of pixels on the image plane equivalent to a distance on a planar surface are described in this section. First, define d as the distance to the target, θ as the VFOV of the camera, ϕ as half of the VFOV (θ), and h as the euclidean distance in meters from the center of the image to the top (and the center to the bottom). The variables are shown in Fig. A.1.

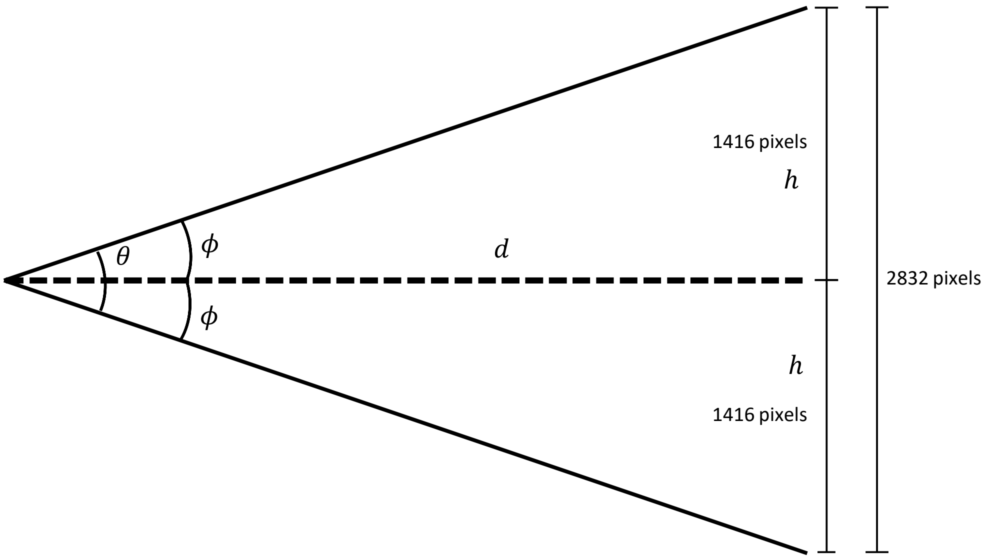


Fig. A.1: Diagram describing the variables needed to determine the number of pixels per unit measure at a given distance d.

To describe the process, an example is provided. When the TC is aligned, the physical offset between the sensors must be known to determine a pixel offset at the alignment distance d. Let the physical offset be 7 cm. First, determine the number of pixels per unit measure (cm in this example) given by (A.5).

$$h = d \times tan(\phi) \tag{A.5}$$

Using $d = 28m$ and $\phi = 14.95°$,

$$h = 28m \times tan(14.95°) = 7.48m \tag{A.6}$$

Now the numbers of pixels per cm ($\frac{pixels}{cm}$) can be computed for the distance h. This is by dividing the number of pixels in spanned by h (1416 pixels) by h, the distance computed using (A.5).

$$\frac{1416}{748}\frac{\text{pixels}}{\text{cm}} = 1.89\frac{\text{pixels}}{\text{cm}} \tag{A.7}$$

Now that the ratio of pixels to distance in cm has been found, the number of pixels equivalent to the physical offset between the camera and lidar sensor can be computed. This is done using the equation below to compute the final solution.

$$7\,\text{cm} \times 1.89\frac{\text{pixels}}{\text{cm}} = 13.23\,\text{pixels} \tag{A.8}$$