AUTOMATIC REGISTRATION OF MULTIPLE TEXEL IMAGES TO FORM A 3-DIMENSIONAL TEXEL IMAGE

by

Neeraj S. Badamikar

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

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

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

UTAH STATE UNIVERSITY
Logan, Utah

2014

Abstract

Automatic Registration of Multiple Texel Images to Form a 3-Dimensional Texel Image

by

Neeraj S. Badamikar, Master of Science

Utah State University, 2014

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

Three-dimensional (3D) imagery has gained a lot of importance in todays world, be it in the field of entertainment, documentation, or defense. Multiple methods for creating 3D images have been proposed in the past. A few famous methods used for 3D image matching are those that include usage of 2D images as stereo pairs or computing 3D rigid body transformations based on range information of points. The Iterative Closest Point algorithm (ICP) and its variants are well known for registration of point clouds, which can be used to create 3D surfaces.

This thesis provides an algorithm, which is a continuation of the work done previously at Utah State University, to create accurate 3D images based on "texel" images obtained from the handheld texel camera built at USU.

The first part of the thesis briefly reviews the structure and working of the handheld texel camera and the technique of creating texel images using the device and calibrating the images to mitigate the effect of lens distortions. A method is then suggested to reduce the errors in the range information in the image caused by walk error and wiggling error and also to compensate for the timing error induced in the individual pixels of the lidar sensor. A way to add a correcting factor to the range information to compensate for any offset in the origin assumed by the sensor and the actual center of perspective (COP) of the sensor

is suggested in the later part of the thesis, thus correcting the images for the inaccuracies caused by the offset.

The second half of the thesis briefly goes over the work previously done on 3D image matching and registration to produce 3D images. A few changes are suggested in some parts of the existing method, which use concepts of epipolar geometry in the RANSAC algorithm and use planar interpolation to accurately obtain the 3D co-ordinates of points from 2D co-ordinates. An iterative solution is proposed to correct erroneously chosen correspondences or reject bad correspondences to improve the rigid body transformation. The transformation thus obtained is used to compute more point matches, which are in turn used to estimate a more accurate least squares solution for the rigid body transformation.

Results show that the calibration techniques and the changes implemented in the point cloud matching algorithm, suggested in this thesis, improve the accuracy of the images and produce 3D images with correct matching.

(91 pages)

Public Abstract

Automatic Registration of Multiple Texel Images to Form a 3-Dimensional Texel Image

by

Neeraj S. Badamikar, Master of Science

Utah State University, 2014

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

Three-dimensional (3D) imaging has found a lot of use in modern era, owing to the fast ongoing advancement in electronics, processors, and sensors, all available at affordable prices. Three-dimensional television, games, and 3D projectors are some of the commonly found utilities which use 3D imaging. 3D images used for surveillance, exploration, target acquisition equipment, etc., are few of the military applications of 3D imaging. As the need increases, more techniques are being introduced to create 3D images which improve upon previously stated techniques. This thesis proposes an algorithm which uses "texel" images, which are images that have digital imagery fused to 3D points, to stitch together a 3D image of an scene.

A brief overview of the methods of calibration developed previously is given, followed by a method to improve the quality of range information obtained from the lidar sensor, which is prone to errors common to the type of sensor used. These errors cause the range measurements to deviate from the true values.

The second half of the thesis briefly goes over the work previously done on 3D image matching to produce 3D images. A few changes are suggested in the method proposed previously. Additional improvement using epipolar geometry, which is a geometric interpretation of 3D projective geometry, helps improve the results obtained. A solution is suggested that

makes use of the transformation computed initially to delete bad or incorrect points from the set of points which are used to compute the initial transformation. A solution obtained from this new set of correct points is more accurate than the one computed initially. This solution is used to generate more points in the point set, which are used to compute the final transformation.

The results show that the methods implemented previously and the ones suggested in this thesis provide accurate 3D images, at low computing cost.

Dedicated to Mr. and Mrs. Dharadhar, my ajoba and aaji...

Acknowledgments

This research project has been an excellent learning curve for me, both academically and personally. Most importantly, I would like to thank my advisor and major professor, Dr. Scott E. Budge, for introducing me to the world of 3D geometry, and trusting my abilities while accepting me in the research group and entrusting me with this research project. His unwaivering support and patience with a novice like me, and his willingness to listen and answer all my questions, are a few of the things I really am thankful for. I would also like to thank him for supporting me financially and for his forgiving nature, especially when the camera was accidently toppled over. Thank you, Dr. Budge.

I would also like to thank Dr. Jacob Gunther and Dr. Don Cripps for being a part of my graduate committee and for their insightful suggestions and help during the course of education at the Utah State University. I would also like to thank the ECE Department for the financial support I received during the initial stages of my master's program.

I would like to thank all my fellow colleagues at the Center of Advanced Imaging LADAR (CAIL), Ziang Wang, Cody Smith, Brittin Bennet, Xuan Xie, and Cody Killpack, for the support and guidance I received during my initial days at CAIL.

Special thanks to my relatives in the USA and my loving family-my aunt, uncle, cousins, mummy aji, my parents, my elder brother and his newlywed wife-for being there when I needed them the most. I owe all my success to my parents for supporting me morally and giving me the most appropriate advice, always. I can never thank them enough for all that they have done for me.

I would like to thank my friends in the USA who have never let me feel separated from my home and family. I am lucky to have found friends in people like Siddharth, Tejas, Ravi, Deepti, Shardul, Satya, Saurabh, Kshitij, David Neal, Andrew Nielson, Ashish, Bhashwati, Bidisha, Ruchir, Rajee, Swathi, Pratima, Sulchan, and many others.

Neeraj Badamikar

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Three-dimensional (3D) Geometry and Imaging Techniques has been a topic of research for many years, for many researchers in different institutions all over the world. Introduction of new applications has given way to rise in interest in 3D geometry and imaging, which in turn has resulted in more number of applications using 3D imagery. Applications like 3D videos, games with 3D views, motion sensors, etc., are few of the applications used for recreation. Applications like 3D imaging of internal body organs, historical sites, etc., have been helpful in proper documentation and studying purposes. Other applications like 3D terrain mapping, automatic target recognition (ATR) have been useful in getting information or surveying of things or places which are not easily accessible for humans. The list goes on. To improve on the performance and accuracy of such applications, much interest is being taken in improving the technology for building 3D images using different techniques.

Multiple techniques have been developed to combine images or other information from multiple sensors at a location to create a 3D surface, with texture information.

A widely used method to extract 3D depth information from images is using stereo vision. A pair of cameras captures an image of the same target from two different views, and the range information is extracted using the common feature points in both the images. A single camera taking images of a scene from two different positions can also be used for stereo matching. A projective transformation is computed using the common points in both the images. This transformation is computed and used to match two images of a scene taken from different poses. From this, 3D points are found using triangulation.

Another method, which is used widely to reconstruct 3D surfaces using multiple distance measurements in a scene, taken from different perspectives, is the Iterative Closest

Point algorithm (ICP). The method is simple and uses sets of 3D points called point clouds to match and create a 3D point cloud combining information from individual point clouds. This method was first introduced by Besl and McKay in 1992 [1]. The method is conceptually straightforward. It minimizes the distance (error) between the points in the two sets and adjusts the transformation, which is in the form of rotation and translation, with each iteration until the error is minimum. Multiple variants of the ICP technique have evolved over the years, as given by Rusinkiewicz and Levoy [2]. Point-to-point error and point-to-plane [3] error are the two types of error functions used commonly when ICP is implemented on point clouds.

Many other techniques have evolved from the basic ones mentioned above. An effort to combine the properties of 2D and 3D image matching techniques mentioned before was made and a method which uses feature-based point cloud matching was introduced by Huang and You [4]. This method uses the property of self-similarity, which is the principle on which 2D image matching algorithms work, on 3D point clouds to overcome the problem of the preprocessing needed when the point clouds are partially overlapped. The sensors used in this method need to be highly accurate in their measurements and noise free for the algorithm to detect accurate correspondences in the two datasets.

A method proposed by Schouweaars *et al.* [5] used multiple visual cameras which were mounted on a road vehicle to form stereo pairs. The range measurements were obtained using a global positioning system (GPS) device installed on the vehicle. Combining the range information and visual information thus obtained, an effort was made to model the street view in a city.

USU has developed a unique solution to create 2.5D images. A 2.5D image can be thought of as 2D image with range information in one plane or along one direction only. A range sensor and an imaging sensor are mounted on a single device in such a way that both sensors view the same target. The information collected from these sensors is combined at the time of image acquisition. The device so formed is called the "Texel Camera." A lidar sensor acts as the range sensor and a simple electro-optic camera is used as the imaging

sensor. The 2.5D image obtained from combining the data received from these sensors is called as the "Texel Image." More information about the texel camera and the 2.5D texel image is given in Chapter 2 of this thesis.

Combining these texel images to generate 3D images was tried by Boldt in his master's thesis [6]. Boldt used the homography technique for calculating the transformation matrix and image registration. Boldt's algorithm is discussed in detail in Chapter 3.

In this thesis, a method has been proposed to modify and improve upon Boldt's algorithm. The method uses concepts of epipolar geometry and projective geometry to detect correct points, and planar interpolation to increase accuracy of the 2D to 3D transformation of points. In addition to this, a few extra iterations have been added to improve upon the correctness of the transformation calculated by eliminating bad points, and also to increase the count of points used in calculating the final transformation, thus making it more robust. This method is discussed in Chapter 3. Results obtained by this method are better in terms of accuracy, within the limitations of the sensors used. The results are presented in Chapter 4 and Chapter 5 concludes the thesis.

Chapter 2

Texel Camera, Texel Images, and Calibration

A 3D image consists of information pertaining to distances of all the points in the image from the sensor, called the range information, and the texture information at all the points in the image as observed from the direction of the sensor. Multiple devices or sensors can be used to get the range information and the texture information, which when merged together, form a 3D image with texture information. A device built at Utah State University has a lidar sensor and an image sensor mounted on it, which acquire the range and texture information required to construct a 3D image. The following section talks about lidar technology and the next section talks about the texel camera built at Utah State University and the components used in it. The last two sections in this chapter describe the texel images and the calibration of the texel camera, respectively.

2.1 Lidar Technology

Lidar is an acronym for LIght Detection And Ranging. It is synonymous with ladar, which stands for LAser Detection And Ranging. For consistency, the term lidar is used in the subsequent sections and chapters of this thesis. Lidar is one of the many methods known to obtain depth information or range information of a real-world object in the view of the sensor [7]. Lidar uses techniques similar to those of a radar. However, lidar operates on much higher frequencies than radar. This restricts the use of lidar to a shorter range, but the range measurements of a lidar are highly accurate.

Lidar is classified mainly into three types, based on how the range is calculated: pulsed time-of-flight, frequency modulated continuous wave and amplitude modulated continuous wave [7]. All the three techniques effectively measure the time-of -flight of the transmitted signal and the range information is derived from it. Pulsed time-of-flight lidar emits a

pulse of light and calculates the range information based on the time taken by the light pulse to return, given that the time taken is linearly dependent on the distance between the sensor and the reflecting surface or the target. The frequency modulated continuous wave sensor transmits a beam which has a varying frequency as a function of time to produce a symmetrical frequency modulated waveform. The echo signals received after reflecting off an object are frequency shifted, as a function of range and Doppler shift, from the signal being transmitted at that time instant. Range information is derived from this frequency shift [8]. The amplitude modulated continuous wave sensor transmits a burst of light signal at a known modulating frequency and phase and measures the time taken for the light pulse to return based on change in the phase of the reflected signal. This time, like in the pulsed time-of-flight sensor, is linearly dependent on the distance of the object from the sensor. Thus the phase offset is measured and used to calculate the range information in an amplitude modulated continuous wave sensor.

Most lidar systems generate a dataset called as point cloud, which is a set of points in 3D space. Usually the points are placed so as to describe the shape, size and distance of the real-world object that is in the view of the lidar sensor. Each point in the point cloud is described with values which describe the position of the point in 3D space, with respect to a known origin, and the power of the light reflected from that point. These values can be termed as range and intensity, respectively. The lidar sensor used in the texel camera assembled at Utah State University is an amplitude modulated continuous wave sensor, which measures range information and internally converts the range into depth information. It is assumed that depth is the distance measured along principle axis (z-axis) of the texel coordinate system.

2.2 Texel Camera

A texel camera is a device which has a lidar sensor and an image sensor mounted on the same device. The lidar sensor provides with the range information and the image sensor adds the texture information of the target scene to generate a texel image. A texel camera was assembled at Utah State University which consists of a lidar sensor and an image sensor,

installed in a way that their optical pathways coincide. This is ensured by the process of coboresighting of the two sensors. Coboresighting reduces the errors introduced due to parallax. If two sensors have parallax, then there is a difference in the angle in which an object is viewed by the two sensors individually [9]. This change in angle between the views of the two sensors varies with distance. Hence, if parallax exists, the two sensors cannot view the same object from the same point of view. This results in erroneous matching of range and texture information obtained by individual sensors. The process of coboresighting is described in the ensuing sections.

The components of the texel camera built at Utah State University are mentioned in the next part of this section. The image of the assembled texel camera is shown in Fig. 2.1.

2.2.1 Lidar Sensor

A Canesta 64 x 64 CMOS TOF lidar was used as the lidar sensor on the texel camera. The 64 x 64 array of sensors give 4096 pixels or points in the point cloud, which are captured simultaneously at multiple frames per second. Since the capture is done simultaneously for all points and the fact that the entire view of the sensor is illuminated by a low-powered light, no mechanical mechanism is needed to facilitate movement of the sensor. The sensor is built on a CMOS chip. Hence, the production of such sensors can be done easily and the cost of the sensor is not high.

Multiple parameters need to be taken into account while using the lidar sensor. These include:

1. Shutter Time,
2. Frame Rate,
3. Common Mode Rejection ratio (CMR),
4. Modulating Frequency,
5. Power of the light which is used for illumination of the scene,
6. Count of Rows and Columns.

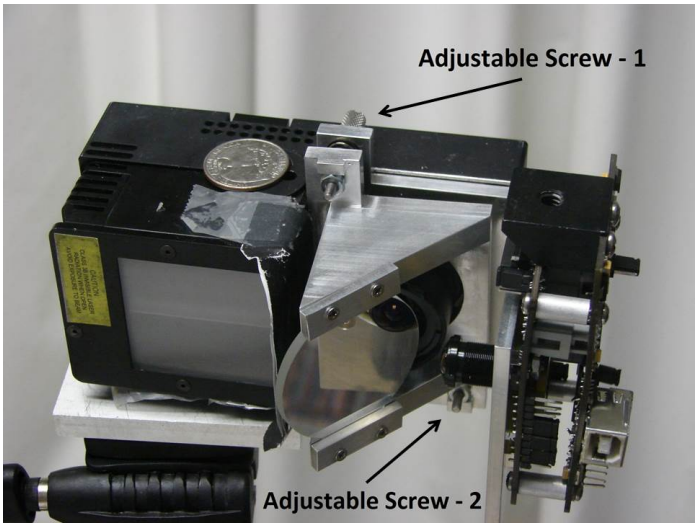


Fig. 2.1: Handheld texel camera.

The characterization of the sensor with respect to the above mentioned parameters is described in Boldt's thesis [6].

2.2.2 Image Sensor

The image sensor used to capture texture information is a Micron 1280 x 1024 CMOS color Electro-Optical sensor, or EO sensor, packaged into a stand-alone development board. The sensor takes a picture and stores it in a RGB image format. Originally the field of view (FOV) of the lidar camera was approximately fifty degree and that of the EO sensor was twenty three degree. As a result, for the two images to match together, the lidar data outside the FOV of the EO camera would be wasted. Hence, to accommodate more EO data, the lens of the EO sensor was changed. The lens chosen for the EO sensor has a focal length of 2.5 mm, and the FOV of the EO sensor is approximately equal to 72 degree which is larger that the FOV of the lidar sensor. For consistency, the EO sensor is referred to as EO camera in this thesis.

2.2.3 Cold Mirror

The texel camera used one of the better ways to merge the range information and the texture information obtained from the respective sensors. The merging was done at the

time of image capture, and in order to facilitate this, it was imperative to ensure that both the sensors had the same view of the target, without any parallax. This meant that the sensors' optical pathways should be coinciding. To ensure this, a cold mirror was used.

A cold mirror reflects light of high frequencies and transmits light of lower frequencies. This property of the cold mirror enables the user to separate the infrared light and the visual light. The visual light is reflected by the cold mirror, while the infrared light is transmitted through it. Thus, the cold mirror is able to separate the incoming light into infrared light, for the lidar sensor, and the visual light, for the EO camera. The positioning of the cold mirror and the two sensors is shown in Fig. 2.2.

Since the source of the lidar signal and the receiver are placed very close to each other, the infrared light emitted by the source leaks through the cold mirror and causes interference with the visual light. To rectify this, a small baffle screen is placed between the lidar source and the cold mirror. The baffle is made up of thick black paper, which absorbs the infrared light incident on it, thus preventing it from interfering with the EO camera. However, introducing a baffle causes less power to be emitted on one side of the system, which in turn results in the power of reflected light to be less on that side, thus causing errors or distortions on one side of the image. The positioning of the baffle with respect to the cold mirror and the two sensors is shown in Fig. 2.3.

2.2.4 Mechanical Mount

A custom mechanical mount was designed to enable positioning of the two sensors at right angle with respect to one another, and the cold mirror at an angle of close to 45 degrees, so as to reflect visual light on the EO camera and transmit the infrared light to the lidar sensor. Adjustable screws are provided on the mount to adjust the position of the cold mirror to fine tune the angle and the position with respect to the two sensors. Two out of the three screw are marked in Fig. 2.1.

2.3 Texel Image

The image obtained by merging the range information received from the lidar sensor

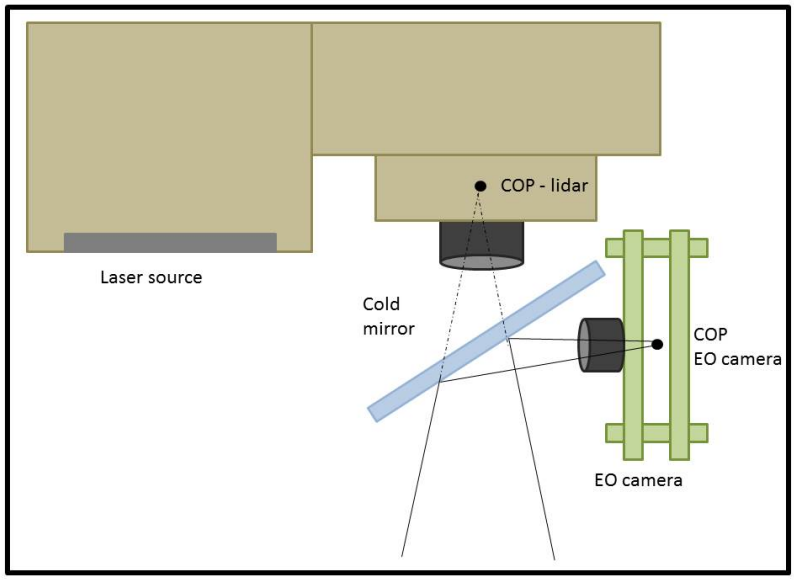


Fig. 2.2: Top view of texel camera without baffle.

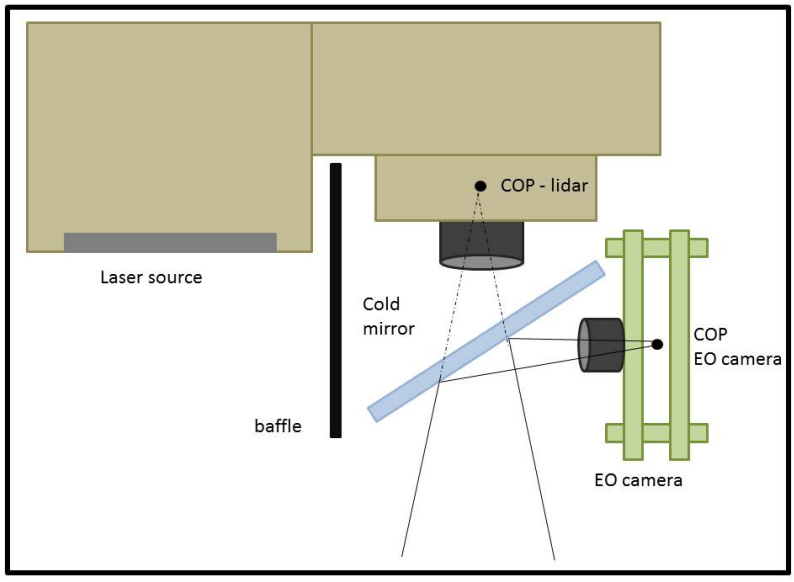


Fig. 2.3: Top view of texel camera with baffle.

and the texture information from the EO camera results in a 2.5D texel image. A 2.5D image can be described as a 3D image viewed from only one point of view. The important difference between a 3D image and a 2.5D image is that in a 2.5D image, range information is available only for the points that are in direct field of view of the sensor. The range information for the points on the target that are obstructed by some part of the target or are not in direct field of view of the sensor is not measured. As a result, such points are not seen in the texel image, which is a 2.5D image.

The lidar sensor measures the range information and provides a depth image and an intensity image as outputs. A point cloud is computed using this information, which gives the x, y, and z coordinate for each pixel or point obtained from the sensor, assuming the origin to be at the COP of the sensor. A mesh or a tessellation of triangles is formed within these points and a wireframe surface or a mesh of triangles is created. The texture information obtained from the EO camera is then superimposed on this triangulated wireframe. Thus the resulting image is a 2.5D image with texture information superimposed on the surface. This merged dataset is called as a texel image.

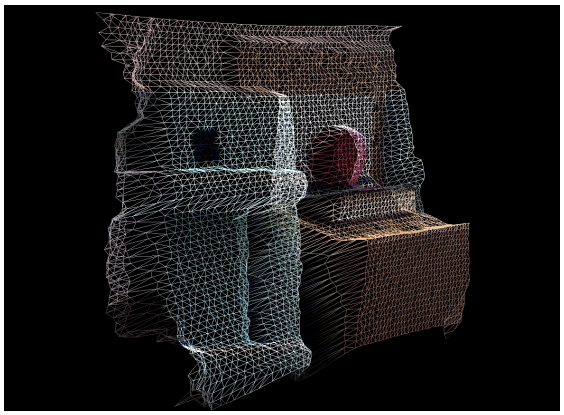
The handheld texel camera built at Utah State University is designed to output texel images of the target in the field of view. The images or datasets obtained from individual sensors and the texel image formed by merging the individual images are shown in Fig. 2.4.

2.4 Calibration

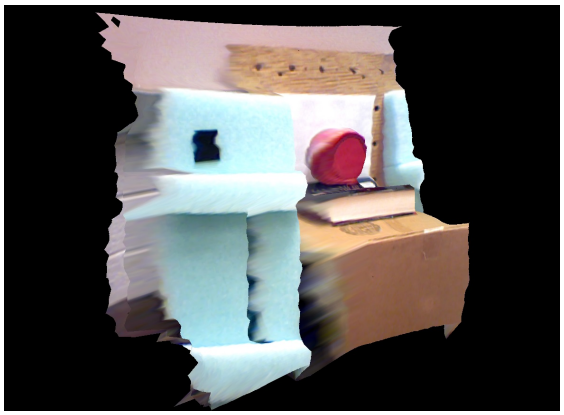
Calibration deals with reducing of any distortion or error caused by the lens of the EO camera or the lidar sensor on the EO image or the depth image, respectively, and then mapping the EO image on the depth image to create an accurate texel image. The steps involved are:

1. Focusing the image sensor, or the EO camera;
2. Coboresighting the EO camera and the lidar sensor;
3. Calibrating individual sensors to mitigate the distortions caused due to the lenses on the sensors (Geometric Calibration);

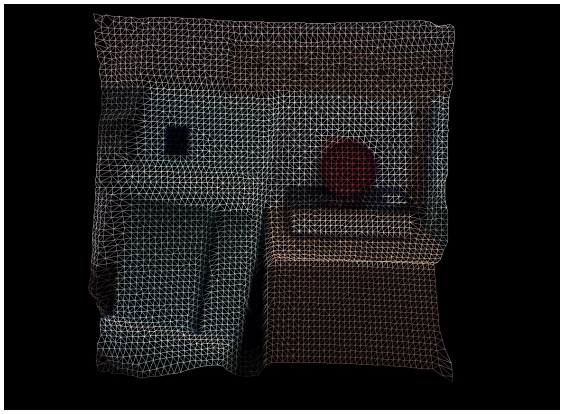
(a)2D EO image



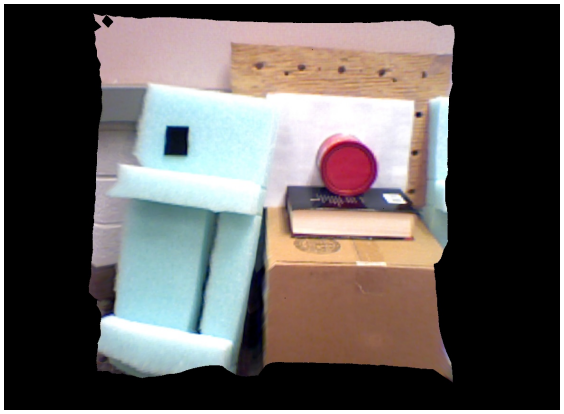
(b) Left hand side view of the triangulated wireframe.



(c) Left hand side view of the 3D texel image.



(d) Front view of the wireframe.



(e) Front view of the texel image.

Fig. 2.4: Texel image.

4. Determine a mapping between the depth image and the EO image;

5. Calibrate for the range imperfections caused due to systematic error in range measurements of the TOF sensor (Range Calibration).

These steps are explained in detail in this section. Of the above mentioned steps, step 1 and step 2 need to be done before the image calibration process begins (step 3 through step 5). Step 1 and 2 involve adjusting the position of the cold mirror using the screws that hold the cold mirror in place. Two out of the three screws can be seen in Fig. 2.1. Once step 1 and step 2 are completed, the mechanical adjustments are completed and the settings must not be disturbed any time in the future. If changed, the entire calibration process needs to be redone. Steps 3 and 4 form the process of "Geometric Calibration" of the texel camera while step 5 deals with the "Range Calibration" of the lidar sensor in particular.

2.4.1 Focusing the EO Camera

It is important to focus the EO camera in order to obtain a sharp EO image of the target in view. The EO camera can be focused by screwing the lens of the EO camera in or out till a sharp image is obtained.

2.4.2 Coboresighting the EO Camera and Lidar Camera

Before the actual calibration, it must be ensured that the fields of view of the two sensors are the same. Since the texture information (the EO image) is obtained from the EO camera, it is of immense importance that the FOV of the EO camera and the lidar sensor coincide. If the light reflected by the cold mirror is not aligned with the principle ray of the EO camera, it causes linear distortion or "shift" in the image, which is not desirable. This distortion can be minimized by adjusting the position and the angle of the cold mirror using the three screws on the mechanical mount that facilitate the rotation of the cold mirror about two axes and translation along the third axis. Also, the effect of parallax in both the sensors needs to be removed.

The point at which the camera can rotate about, without suffering a parallax effect is called the center of perspective (COP) [10, 11]. In order to minimize the parallax between both the sensors on the texel camera, the individual COPs of the sensors should coincide optically. The position of the mirror is adjusted until the principle axes of the two sensors are coaxial and the COP of the lidar sensor coincides with the virtual COP of the EO camera. This procedure can be carried out using a panoramic mount, which helps in finding out the COP of a camera. The panoramic mount is a device that can be mounted on a tripod, and is used to capture scenes that can be stitched together to form a panorama. It allows the camera to rotate about vertical and horizontal axes, which can be measured relative to the camera.

The process of coboresighting is done in four steps [12]. Also note that it is convenient to use the "brightness image" that is obtained from the lidar sensor, than using the depth image.

The steps involved for coboresighting the two sensors are:

1. **Match the fields of view**: The position and the angle of the cold mirror is adjusted such that the center of view of the EO camera matches the center of field of view of the lidar sensor.

2. **Find COP of both the EO camera and lidar sensor**: The texel camera is mounted on the panoramic mount to first find the COP of lidar. Since the mechanical mount is designed in a way to position the cold mirror and the two sensors such that the virtual COP of the EO camera is located at the COP of the lidar, both the sensors can be checked simultaneously. When the COP of the lidar is found in either of the two axes, rotation about that axis will cause no relative shift in objects closer to the sensor with respect to objects far away in the scene. If the virtual COP of the EO camera is also located at the same point, there will be no shift in objects closer and further away in the EO image during rotation.

3. **Re-adjust the cold mirror**: The position of the cold mirror must be adjusted, by means of the three screws on the mechanical mount, so that the virtual COP of the

EO camera coincides with the COP of the lidar sensor.

4. **Repeat**: Iterate in steps 1 - 3 till the COPs of the sensors coincide.

Once the process of coboresighting is completed, the screws on the mechanical mount should be fixed to their positions and must not be re-adjusted, as this would result in shifting of the COPs of the two sensor with respect to each other. The camera is now ready to be calibrated. The calibration process further can be divided into two major steps. The steps consist of geometric calibration, applicable to both sensors, and range calibration which is applicable to the lidar sensor.

2.4.3 Geometric Calibration

Geometric calibration minimizes errors and distortions in the images which are caused due to the lens or the cold mirror. Since both the sensors have different lenses, one approach would be to calibrate them individually and then find a mapping between them. A method was chosen to calibrate the lidar sensor first, and then a mapping was found from the EO image to the calibrated lidar image, so that the EO image gets calibrated through the mapping [12].

2.4.3.1 Calibrating the Lidar Sensor

A point in 3D space can be defined as $\boldsymbol{P}_r = [X_r, Y_r, Z_r]^T$, where Z_r is the distance measured along the principal axis of the camera coordinate system.

The principal axis is the line passing through the center of curvature of the lens and is normal to the plane tangential to the lens. Here we assume the principal point, which is the point at which the principal axis intersects the sensor array, as the origin $[0, 0, 0]$ in a Cartesian coordinate system where x-axis is parallel to the rows of the sensor array, y-axis is parallel to the columns of the sensor array and the z-axis is normal to the sensor array. The object is assumed to be in $-z$ direction, meaning the z coordinate of a point on the target decreases as it goes further from the principal point, along the principal axis.

Comparing it to the ideal pinhole camera model [13], the projected coordinates of the the 3D point into 2D space are given by

$$\boldsymbol{p}_n = \begin{bmatrix} \frac{X_r}{Z_r} \\ \frac{Y_r}{Z_r} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix}. \tag{2.1}$$

Here $\boldsymbol{p_n}$ is called as the normalized coordinate of the point $\boldsymbol{P}_r$.

However, owing to the distortions introduced due to the lens, the normalized point is changed. The distorted point can be defined as

$$\boldsymbol{p}_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix} = d_r \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \boldsymbol{d}_t, \tag{2.2}$$

where d_r is due to the radial distortion and $\boldsymbol{d}_t$ is due to the tangential distortion. We can define these by

$$d_r = 1 + k_1 r^2 + k_2 r^4 + k_5 r^6, \tag{2.3}$$

$$\boldsymbol{d}_t = \begin{bmatrix} 2k_3 x_n y_n + k_4(r^2 + 2x_n^2) \\ k_3(r^2 + 2y_n^2) + 2k_4 x_n y_n \end{bmatrix}, \tag{2.4}$$

and

$$r^2 = x_n^2 + y_n^2. \tag{2.5}$$

This definition is according to the model of camera distortion given by Heikkila and Silven [13]. The coordinates of the point on the sensor array can be determined using the

distorted points $\boldsymbol{p}_d$ and the camera calibration parameters using

$$\boldsymbol{p}_p = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}, \tag{2.6}$$

where $\boldsymbol{p}_p$ is the pixel position on the sensor array, given in pixel coordinates, and $\boldsymbol{K}$ is the camera matrix containing the intrinsic parameters,

$$\boldsymbol{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.7}$$

Here f_x and f_y are the focal distances in the x and y direction, s is the skew factor, c_x and c_y are the horizontal and vertical position of the principal point. All these parameters are defined in pixel coordinates. The principal point is defined with respect to the origin of the pixel matrix, in which the origin is at the top-left corner. The intrinsic parameters or the $\boldsymbol{K}$ matrix is calculated using the Camera Calibration Toolbox in Matlab [14]. The procedure for finding these parameters along with the distortion coefficients is described by Boldt [6]. These camera parameters are useful in transforming a point on the sensor array $\boldsymbol{p}_p$ to distorted point coordinates $\boldsymbol{p}_d$. These parameters are constant for a fixed optical setup, and need to be calculated only once. However, if the position of the mirror or the lens is disturbed, these values need to be recomputed.

2.4.3.2 Computing the Normalized Image Coordinates

The distorted point locations $\boldsymbol{p}_d$ and the distortion parameters d_r and $\boldsymbol{d}_t$ can be used to transform points from the sensor array $\boldsymbol{p}_p$ to points in normalized space $\boldsymbol{p}_n$. This is done by first inverting the camera matrix $\boldsymbol{K}$, and obtaining distorted point locations $\boldsymbol{p}_d$. Using

the mapping defined for the distortion parameters d_r and $\boldsymbol{d}_t$ in (2.4) and (2.5), given as

$$x_d(x_n, y_n) = x_n + k_1\, x_n r^2 + k_2\, x_n r^4 + k_5\, x_n r^6 + 2\, k_3\, x_n\, y_n + k_4(r^2 + 2\, x_n^2), \tag{2.8}$$

$$y_d(x_n, y_n) = y_n + k_1\, y_n r^2 + k_2\, y_n r^4 + k_5\, y_n r^6 + k_3(r^2 + 2\, y_n^2) + 2\, k_4\, x_n\, y_n. \tag{2.9}$$

From this, an approximate solution for $\boldsymbol{p}_n(x_n, y_n)$ can be found, given as $\tilde{\boldsymbol{p}}_n(\tilde{x}_n, \tilde{y}_n)$, using the approach given by Melen [15].

The normalized values $\tilde{\boldsymbol{p}}_n$ are fixed for each pixel, given that the calibration matrix and the distortion parameters are unchanged. Hence, the $\tilde{\boldsymbol{p}}_n$ values can be computed once and stored in a look up table, and can be used when needed. The $\tilde{\boldsymbol{p}}_n$ values are important in transforming the points on sensor array into 3D space.

2.4.3.3 Transforming 2D Points on Lidar Sensor Array into 3D Space

Equations (2.1)-(2.6) describe how to map a 3D point in space onto the lidar sensor array. The location of the point in space using 3D coordinates (x,y,z), can be computed based on the measurements of the lidar sensor. The lidar sensor measures the range of the point. In addition to this, only the pixel points $\boldsymbol{p}_p$ are known. Using this information, the location of the point in 3D space $\tilde{\boldsymbol{P}}_r$ can be computed as follows.

As mentioned before, the normalized points $\tilde{\boldsymbol{p}}_n$ are computed approximately by inverting the camera matrix $\boldsymbol{K}$ and the expressions given in (2.8) and (2.9). Once the $\tilde{\boldsymbol{p}}_n$ are known, the 3D point $\tilde{\boldsymbol{P}}_r$ can be computed using the range λ_r.

$$\tilde{\boldsymbol{P}}_r = \frac{\lambda_r}{(\tilde{x}_n^2 + \tilde{y}_n^2 + 1)^{\frac{1}{2}}} \begin{bmatrix} \tilde{x}_n \\ \tilde{y}_n \\ 1 \end{bmatrix} = \lambda_r \begin{bmatrix} \tilde{x}_n z_c \\ \tilde{y}_n z_c \\ z_c \end{bmatrix} = \lambda_r \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}, \tag{2.10}$$

where

$$z_c = (\tilde{x}_n^2 + \tilde{y}_n^2 + 1)^{-\frac{1}{2}}. \tag{2.11}$$

2.4.3.4 Lidar to Visual Image Transformation

As mentioned before, both the sensors on the texel camera have different optical systems and the cold mirror is positioned to enable both the sensors to view the same scene. However, the position of the cold mirror is still prone to shifts and errors. Also, the fields of view of both the sensors are different, the EO camera having a larger field of view than the lidar sensor. Hence, a mapping function needs to be formed, in order to match the two sensors so that they view the same scene. The procedure to find and use the mapping function is described by Budge and Badamikar [12].

Since the EO camera is also subject to distortion caused by the lens, the mapping function should be such that it minimizes the abnormalities caused by the lens. The mapping function is based on the camera model described in equations (2.8) and (2.9). The mapping is found between the pixels in normalized space $\boldsymbol{p}_n$ and the corresponding points in EO image (u,v). Since the lidar pixels in normalized space have already been corrected for radial and tangential distortion, the imperfections added by the lidar sensor lens are removed. In order to correct the EO image, an unknown camera matrix $\boldsymbol{K^I}$ is applied to the camera model for lidar sensor. The mapping then can be given as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boldsymbol{K^I} \begin{bmatrix} \tilde{x}_d(\tilde{x}_n, \tilde{y}_n) \\ \tilde{y}_d(\tilde{x}_n, \tilde{y}_n) \\ 1 \end{bmatrix} = \begin{bmatrix} f_x^I \tilde{x}_d(\tilde{x}_n, \tilde{y}_n) + s^I \tilde{y}_d(\tilde{x}_n, \tilde{y}_n) + c_x^I \\ f_y^I \tilde{y}_d(\tilde{x}_n, \tilde{y}_n) + c_y^I \\ 1 \end{bmatrix}. \tag{2.12}$$

After combining constants, the resulting mapping is given by

$$\begin{aligned} u = {} & g_1\,\tilde{x}_n + g_2\,\tilde{x}_n r^2 + g_3\,\tilde{x}_n r^4 + g_4\,\tilde{x}_n r^6 + g_5\,\tilde{x}_n\,\tilde{y}_n + g_6(r^2 + 2\,\tilde{x}_n^2) + g_7\,\tilde{y}_n + \\ & g_8\,\tilde{y}_n r^2 + g_9\,\tilde{y}_n r^4 + g_{10}\,\tilde{y}_n r^6 + g_{11}(r^2 + 2\,\tilde{y}_n^2) + g_{12} \\ v = {} & h_1\,\tilde{y}_n + h_2\,\tilde{y}_n r^2 + h_3\,\tilde{y}_n r^4 + h_4\,\tilde{y}_n r^6 + h_5(r^2 + 2\,\tilde{y}_n^2) + h_6\,\tilde{x}_n\,\tilde{y}_n + h_7. \end{aligned} \tag{2.13}$$

The high order r^6 term in this polynomial can lead to an ill-conditioned solution. Secondly, it would be unreasonable to assume that the radial distortion is strictly circular, because the imperfect positioning of the cold mirror may cause the radial distortion to be

slightly eccentric. Hence, a more appropriate mapping is derived by removing the higher order terms and the coefficients g_4, g_{10} and h_4, and decoupling terms to express the mapping as a function on x_n and y_n, we get

$$\begin{aligned} u = {} & g_1 + g_2\tilde{x}_n + g_3\tilde{x}_n^2 + g_4\tilde{x}_n^3 + g_5\tilde{y}_n + g_6\tilde{x}_n\tilde{y}_n + g_7\tilde{y}_n^2 + g_8\tilde{y}_n^3 + g_9\tilde{x}_n^2\tilde{y}_n + \\ & g_{10}\tilde{x}_n\tilde{y}_n^2 + g_{11}\tilde{x}_n^5 + g_{12}\tilde{y}_n^5 + g_{13}\tilde{x}_n\tilde{y}_n^4 + g_{14}\tilde{x}_n^4\tilde{y}_n + g_{15}\tilde{x}_n^3\tilde{y}_n^2 + g_{16}\tilde{x}_n^2\tilde{y}_n^3 \\ v = {} & h_1 + h_2\tilde{y}_n + h_3\tilde{y}_n^2 + h_4\tilde{y}_n^3 + h_5\tilde{y}_n\tilde{x}_n + h_6\tilde{x}_n^2 + h_7\tilde{y}_n\tilde{x}_n^2 + h_8\tilde{y}_n^5 + h_9\tilde{y}_n\tilde{x}_n^4 + \\ & h_{10}\tilde{y}_n^3\tilde{x}_n^2. \end{aligned} \tag{2.14}$$

Finally, since the skew parameter s is negligible for nearly all focal planes, and to allow for rotation between the lidar and EO sensors, the mapping can be reduced to

$$\begin{aligned} u = {} & g_1 + g_2\tilde{x}_n + g_3\tilde{x}_n^2 + g_4\tilde{x}_n^3 + g_5\tilde{y}_n + g_6\tilde{x}_n\tilde{y}_n + g_7\tilde{y}_n^2 + g_8\tilde{x}_n\tilde{y}_n^2 + g_9\tilde{x}_n^5 + \\ & g_{10}\tilde{x}_n\tilde{y}_n^4 + g_{11}\tilde{x}_n^3\tilde{y}_n^2 \\ v = {} & h_1 + h_2\tilde{y}_n + h_3\tilde{y}_n^2 + h_4\tilde{y}_n^3 + h_5\tilde{x}_n + h_6\tilde{y}_n\tilde{x}_n + h_7\tilde{x}_n^2 + h_8\tilde{y}_n\tilde{x}_n^2 + h_9\tilde{y}_n^5 + \\ & h_{10}\tilde{y}_n\tilde{x}_n^4 + h_{11}\tilde{y}_n^3\tilde{x}_n^2. \end{aligned} \tag{2.15}$$

To formulate the mapping, the coefficients $\boldsymbol{g}$ and $\boldsymbol{h}$ need to be estimated. Vectors $\boldsymbol{g}$ and $\boldsymbol{h}$ represent the mapping coefficients given in (2.15). These are estimated using multiple corresponding points in the EO image and the brightness image ($\boldsymbol{x}_i \leftrightarrow \tilde{\boldsymbol{p}}_{ni}$), where $\boldsymbol{x}_i$ represents 2D points selected from the EO image and $\tilde{\boldsymbol{p}}_{ni}$ represents normalized coordinates of points in the brightness image obtained from lidar sensor. The corresponding points are obtained by using a checkerboard, with clear and individually distinguishable corners, as a target. Multiple images of the checkerboard can be taken in order to pick points in the entire field of view of the lidar sensor. The points can be picked manually or automatically by using a corner-finder algorithm, which is a part of the Camera Calibration Toolbox in Matlab. Once sufficient number of points in lidar brightness image and their corresponding

points in the EO image are picked, the mapping coefficients can be estimated by a maximum-likelihood fit to the points, as given by

$$(\hat{\boldsymbol{g}}, \hat{\boldsymbol{h}}) = \arg\min_{(\boldsymbol{g},\boldsymbol{h})} \sum_{i=0}^{N-1} d(\tilde{\boldsymbol{p}}_{ni}, \hat{\tilde{\boldsymbol{p}}}_{ni})^2 \frac{\sigma_{EO}}{\sigma_{lidar}} + d(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i)^2, \qquad \hat{\boldsymbol{x}} = \begin{bmatrix} \hat{u} \\ \\ \hat{v} \end{bmatrix} = F(\hat{\tilde{\boldsymbol{p}}}_n, \boldsymbol{g}, \boldsymbol{h}), \quad (2.16)$$

where $\hat{\tilde{\boldsymbol{p}}}_{ni}$ and $\hat{\boldsymbol{x}}_i$ are the estimated positions of the correct (noiseless) observations, $\frac{\sigma_{EO}}{\sigma_{lidar}}$ is the ratio of EO image pixel size to lidar pixel size, $d(\cdot)$ is the Euclidean distance, and $F(\tilde{\boldsymbol{p}}_n, \boldsymbol{g}, \boldsymbol{h})$ is given by (2.15) [16].

2.4.4 Range Calibration

The lidar sensor obtains two types of information for each image that it takes. Firstly, the brightness information, which is the data about the intensity of the light reflected back from the target; and secondly, the depth image, which is derived from the range information, or the distance to the object from the sensor. The range information that is obtained is not highly accurate and is prone to some errors due to imperfections in the sensor. These errors can be broadly classified into two types, a systematic error caused due to non-ideal modulating function, also called the "wiggling error," and an error caused by difference in the amount of light reflected off the surface, known as "walk error." A detailed explanation for the wiggling error is given by Frank *et al.* [17]. Walk error is observed between objects with contrasting reflectivities, and the camera needs to be calibrated to reduce the error caused by this. An effective method to reduce this error is to create a 2D calibration surface, which is a function of the reflected intensity and range. Additional studies based on the location of pixel in the array, in order to account for vignetting, is given by Lindner and Kolb [18]. However, since real-time creation and calibration of texel images was the aim of this effort, a simpler three step approach was attempted.

2.4.4.1 Flat-Field Correction

It was observed that in the lidar sensor, there are slight differences in the location of depth origin for each of the 64 x 64 pixels. This means that, for a flat target with the sensor positioned such that the principle-axis (*z-axis*) is normal to the flat target, the pixels in the image were not exactly flat. The mesh formed by the 3D points in the texel image was not flat as it should have been. This is caused due to the timing and response difference experienced by each pixel. It was assumed that the sensor associated with each of the pixel in the array had a slight timing offset with respect to each other due to the aging or the latency associated with the electronic circuit of each sensor.

In order to reduce the effect of the timing imperfections associated with each pixel, a flat target was used and the distance along the *z-axis* of the sensor was measured, with respect to a point of reference on the sensor. A correction value for each pixel was generated by calculating the difference between the distance measured physically and the distance measured by the sensor. The same procedure was carried out for different depths, and a maximum-likelihood correction for each pixel was computed, over all the depths. The correction was converted to range, and stored in a look-up table (LUT). The corrections are added to the measured range for each pixel in the first step of range calibration. This method is similar to the one given by Kahlmann *et al.* [19] in the effort to correct the range data for fixed pattern noise.

This step in the process of range calibration corrects for the timing mismatch in the sensors for each pixel, such that each pixel is corrected with respect to the reference point on the sensor. This is analogous to a flat-field correction for each pixel, over measured depths.

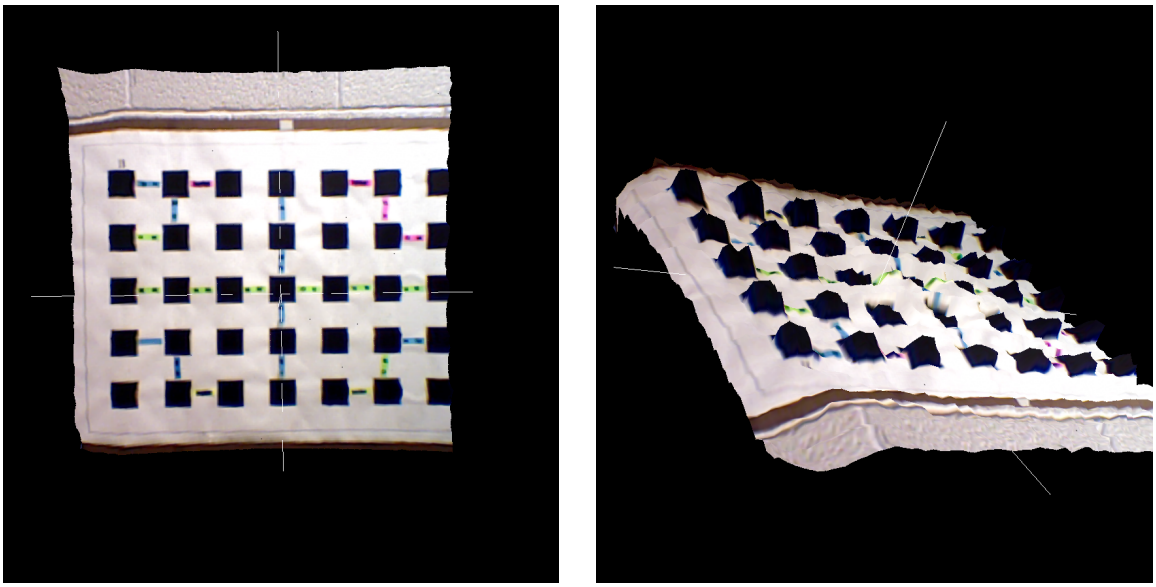
2.4.4.2 Range Error Correction for Walk Error and Wiggling Error

As mentioned before, the walk error results from differences in the target reflectivity and wiggling error results from the imperfections in the modulating signal transmitted by the lidar source. Since the signal is common for all pixels in the lidar sensor array, the correction for the wiggling error or the walk error can be assumed to be the same for all

pixels, unlike the flat-field correction, which was pixel dependent.

It is observed that the error in range is significantly dependent on two things. The magnitude of the error differs according to the intensity of the light reflected from the target object. It is observed that the reflectivity depends on the colors on the surface in the sense that the IR light often reflects better off surfaces that are brightly colored than those with dark colors. For an object with highly contrasting surface colors like a checkerboard pattern on a flat wall, it is observed that the range obtained for pixels corresponding to black squares is higher in magnitude than that for pixels corresponding to white squares. This can be seen in Fig. 2.5. The pixels in the dark squares are projected backward. Figure 2.5(b) shows the texel image tilted in a way that the back of the texel image is seen. The significant difference in range measured for the dark pixels and the range for the white pixels can be seen.

The range error also depends on the actual range of the object. The resulting error is dependent on range, forming a "wiggle pattern."



(a) Flat view of the chequered board texel image

(b) Tilted view of the chequered board texel image

Fig. 2.5: Range error: "walk error" is evident in Fig. 2.5(b) which shows the back view of the target. The dark squares are projected backwards, compared to white areas.

A simple yet effective method to correct the range to compensate for wiggling error is to create a LUT based on correction values for different values of range and intensities. As was suggested before, since the error is dependent on range and intensity values, a 2D correction-surface can be created, which is a function of brightness or the intensity value of a pixel, and the measured range for the pixel. This surface is called a "calibration surface."

A calibration surface was created using a flat target and ensuring that the focal plane of the lidar sensor was parallel to the target (the *z-axis* normal to the target). The range information and the brightness (intensity) information obtained from the sensor for each pixel were recorded. A correction value for range was computed, based on the range given by the sensor and the range measured physically from a reference point on the lidar sensor, called the "true range." This reference point is the same point which was used for reference in the flat-field correction. Thus a correction value for all pixels with varying range and intensity measurements were obtained.

This experiment was repeated over a set of ranges from the target, and for each range the power of the light emitted was varied from maximum value, which avoided saturation of pixels, to a lower value, which ensured at least 75 percent of the pixels were above a threshold intensity of 14 units. The threshold value of 14 was chosen because range measurements for pixels with intensity below 14 units have large error. Thus a dataset was obtained which had data for each pixel with correction values for a wide number of ranges and brightness (intensities).

This data set was quantized with respect to range and intensity values and the correction values were averaged for each cell. The cell size is based on a trade-off between the size of LUT and the accuracy of the correction.

This process of computing the calibration surface is given by Budge and Badamikar [12] and the calibration surface thus computed is shown in Fig. 2.6.

In Fig.2.6, the *z-axis* represents the correction values for the range with respect to the measured range and intensity values, *x-axis* and *y-axis*, respectively. The "wiggling" nature of the error, and hence the correction can be observed in Fig. 2.6.

To calibrate the range measurement $\lambda_r(m,n)$, the range and brightness ($B(m,n)$) are used as input for the LUT, and correction value for range is obtained by interpolation between the closest points known in the LUT. The correction is then added to the range to get the calibrated range $\lambda_r^{cal}(m,n)$.

2.4.4.3 COP Offset Error

There is a possibility that the origin for the coordinate system assumed by the lidar sensor for processing and calculating range information might differ from the COP of the sensor. This can cause an error in the back-projection of 3D points from the sensor array as the back-projection value depends on the measured range, as mentioned in section 2.4.3.3.

For a geometrically calibrated sensor, a COP with a positive Z_r value (the origin assumed by the lidar sensor lies in between the COP and the target) will cause all the computed 3D points to be closer to each other than the actual distance. This can be shown in Fig. 2.7.

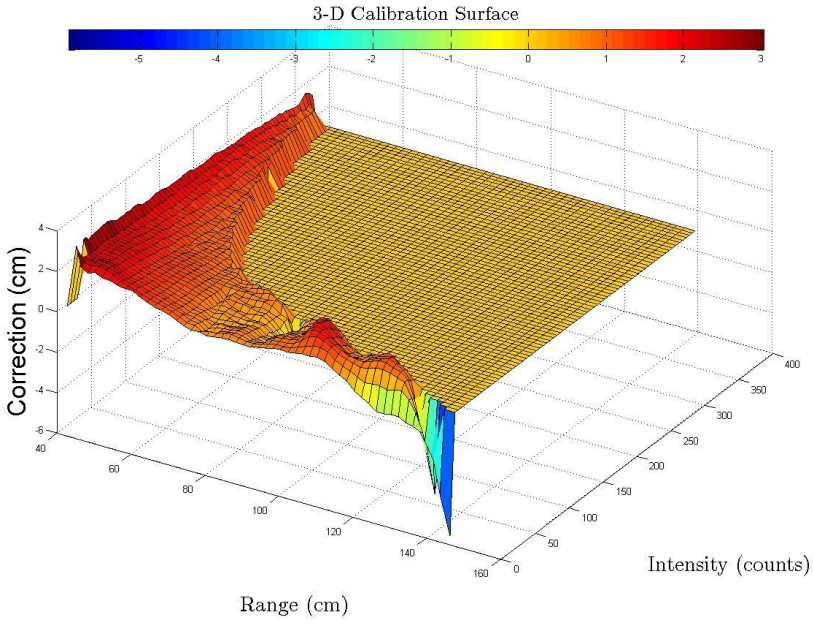


Fig. 2.6: Calibration surface: shows the correction values (z) corresponding to given intensity or brightness (x) and range (y).

In this figure, it can be seen that the COP of the lidar sensor has a positive z-coordinate as compared to the origin assumed by the sensor. The range for a point in 3D space can be computed from depth and normalized coefficients of points on sensor array using (2.10) and (2.11). Since the origin assumed by the lidar sensor is closer to the target along the z-axis, the range $\tilde{\lambda_r}$ measured by the sensor is smaller than the true range λ_r. As a result, the depth measured by the sensor is less than the actual depth of the 3D point on the target. This, in turn, results in the x-coordinate of the point in 3D space, computed using (2.10) and (2.11) to be different than the true value. It can be seen in Fig. 2.7 that since $\tilde{\lambda_r}$ is less than λ_r, the point A' back-projected in 3D space has a x-coordinate lower in magnitude than the true value (shown by point A), such that it is closer to point "B." As a result, the distance $d(A', B')$ is less than the true distance $d(A, B)$.

Similarly, Fig. 2.8 shows the case where the COP of the lidar sensor has a negative z-coordinate as compared to the origin assumed by the sensor. The resulting range measured by the sensor $\tilde{\lambda_r}$ is larger than the true range λ_r, resulting the x-coordinate of the point "A" computed using (2.10) and (2.11) to be different than the true value. It differs such that the x-coordinate of A' is larger than the true value and the back-projected point A' moves away from the point "B." Similarly, coordinates of point B are computed such that it is back-projected as B' which away from point A. As a result the distance $d(A', B')$ appears larger than the true distance $d(A, B)$.

Figures 2.7 and 2.8 show how the offset between the COP of the lidar sensor and the origin assumed by it affects back-projection of points by showing the variation in x-coordinate of the projected point. Similar logic can be applied to see the change in the y-coordinate caused due the offset. As a result, the overall distance between the points in 3D space differs from the true distance.

To alleviate the error caused due to the COP offset, an offset z_o is assumed to be the difference in the z-coordinate in the actual COP and the assumed origin. The optimal value of z_o is calculated such that the distances between 3D points obtained by the sensor are equal to the true distances between the corresponding points in real world. Since the true

value of z_o is unknown, only the (X_r,Y_r) values are used to estimate z_o. The process to estimate the optimum z_o, given by Budge and Badamikar [12], is as follows.

A checkerboard pattern is taken as a target scene, and the 2D points corresponding to corners are measured and defined as $\boldsymbol{P}_r^{m,k}, k = 0, \ldots, K$, for a given depth. K is the number of points in an image taken when the target is at a particular depth from the sensor. Such images are taken over a range of depths $l = 0, \ldots L$, L being the number of different depths at which images were taken, ensuring that the flat target is parallel to the focal plane of lidar sensor. The 2D points computed from the sensor measurements can then be defined as

$$\tilde{\boldsymbol{P}}_r^{l,k} = \begin{bmatrix} x_c^k \lambda_r^{l,k} + z_o \tilde{x}_n^k \\ y_c^k \lambda_r^{l,k} + z_o \tilde{y}_n^k \end{bmatrix}, \tag{2.17}$$

and the optimal value of z_o is computed by minimizing the error between the measured points and true distance, over all the depths. It is given as

$$z_o^{opt} = \arg\min_{z_o} \sum_{l=0}^{L} \sum_{j=0}^{K_l} \sum_{k=j+1}^{K_l} \left[d(\tilde{\boldsymbol{P}}_r^{l,j}, \tilde{\boldsymbol{P}}_r^{l,k}) - d(\boldsymbol{P}_r^{m,j}, \boldsymbol{P}_r^{m,k}) \right]^2. \tag{2.18}$$

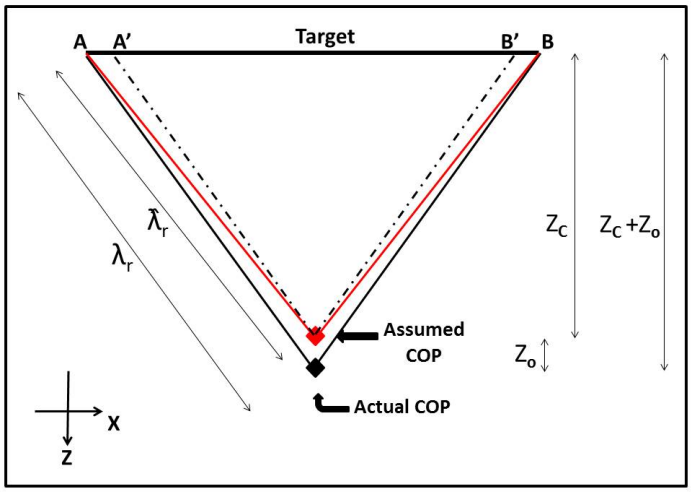


Fig. 2.7: A ray diagram showing the z_o correction: for COP having a positive z- coordinate compared to origin assumed by lidar sensor.

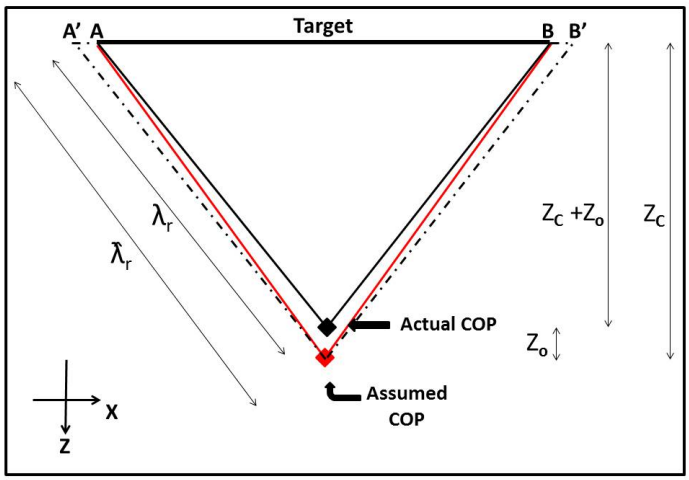


Fig. 2.8: A ray diagram showing the z_o correction: for COP having a negative z- coordinate compared to origin assumed by lidar sensor.

Thus the final calibrated 3D point is given by

$$\tilde{\boldsymbol{P}}_r(m,n) = \begin{bmatrix} X_r^{cal}(m,n) \\ Y_r^{cal}(m,n) \\ Z_r^{cal}(m,n) \end{bmatrix} = \begin{bmatrix} x_c(m,n)\lambda_r^{cal}(m,n) + z_o\tilde{x}_n(m,n) \\ y_c(m,n)\lambda_r^{cal}(m,n) + z_o\tilde{y}_n(m,n) \\ z_c(m,n)\lambda_r^{cal}(m,n) + z_o \end{bmatrix}. \tag{2.19}$$

Chapter 3

Point Cloud Matching and Image Registration

As mentioned previously, the list of applications which use 3D imaging is endless and more applications are being thought of as more accurate and "real" 3D images are generated. Multiple techniques have been developed over the years and efforts are being taken in order to improve the performance of existing techniques and develop new ones which perform better and yield good results. A common technique to create 3D images includes feature-based image registration using projective geometry on 2D images captured from different poses. Two-dimensional images of an object can be used to create a 3D image of the object using stereo methods [20]. Stereo based image matching techniques compute depth information, given the disparity between the COPs of the sensors used to capture the images. A segment based stereo matching algorithm is given by Medoni and Nevatia [21].

However, image matching using 2D images has some disadvantages due to the environment in which images are captured, and also because range information for the target is not measured directly. It is difficult to compute the depth information of a target if the disparity between the two positions from where the images are captured is unknown. Also, finding adequate and accurate features in areas with low light intensity is a tedious task. The issue of difference in perspective between the poses of the camera also causes difficulty in determining corresponding pairs of feature points in the images. Difference in perspective causes the feature matching based on correlation to work poorly, because of the shift in shape of the projection of an object on two image planes, when the images are captured from two different positions. This problem is now being assessed and a few techniques like modifying the shape of correlation window to account for the difference in perspective have been suggested by Tuytelaars and van Gool [22] and Kanade [23]. Multiple other variations of this method, depending on the applications, are also being used widely. However, the

problem of extracting depth information at low-detailed areas or dim light still persists.

Since no range information is available on the points in the image, the issue of scaling makes it impossible to guess the actual range and size of the objects in view. This means that it is impossible to differentiate between an object which is big in size and held at a large distance, and a similar object which is smaller in size but held close to the observer. The issue of scaling can be resolved if range information such as the distance between points on the object and the observer, and the distance between individual points on the object are known [16]. This range information can be obtained using lidar sensors. Lidar sensors, as described in Chapter 2, can be used to measure range information to the target. The lidar sensor generates a dataset of points with range information for each of the points. Such datasets are known as point clouds. An algorithm that merges multiple point clouds that are captured from different perspectives was developed by Besl and McKay, known as the iterative closest point method (ICP) [1]. ICP method computes the transformation by minimizing the overall distance (error) between corresponding points in the two point clouds. However, this technique fails when the two point clouds to be matched are only partially overlapped. Since no information about the amount of overlap is known, often the error converges incorrectly, giving a poor solution. Preprocessing the data sets such that the information of the overlap is provided, so that the error is computed and minimized only over the region of overlap can help address this issue. A method suggestion by Chetverikov *et al.* [24] addresses this issue to a certain extent. A few variations of ICP have been introduced, as per requirements of the application in which it is used [4, 5].

The handheld texel camera developed in USU generates texel images which have texture information, which is similar to 2D texture images, imprinted on wiremesh formed by points whose range information is known. These points form a point cloud which is similar to the dataset used in ICP. The method used to generate and calibrate the texel images is given in Chapter 2. A texel image is a 2.5D image, which can be described as a 3D image viewed from one point of view. The range information and the texture information is available only for the points that are in direct field of view of the sensor. The points that are not in the

field of view of the sensor are not seen in the texel image. Hence, texel images captured from multiple perspectives can be merged to form a complete 3D image of an object. Moreover, since the texel image contains 2D texture data an 3D range information for lidar points, both 2D and 3D image matching methods can be employed to merge the datasets in a better way. This merging of multiple images is called image registration, and a method for registering multiple texel images based on principles of projective geometry and epipolar geometry is proposed in this chapter of the thesis.

3.1 Previous Work Done on Texel Image Matching at USU

Some previous work has been done on image registration to match two texel images of a scene, captured at different locations, using visual and range information. This thesis extends the work previously done by Boldt [6] and Nelson [25]. This section gives a brief background of the methods developed previously. Detailed information of the work can be found in the MS thesis "Point Cloud Matching with a Handheld Texel Camera" by Boldt and "Image-Based Correction of Ladar Pointing Estimates to Improve Merging of Multiple Ladar Point Clouds" by Nelson. This method, along with some variation to improve the performance is given in the work done by Budge and Badamikar [26].

The final aim being matching two texel images, the process followed to achieve desired results can be summarized as:

1. Detect Harris features,
2. Find putative correspondences,
3. Establish a model that fits the corresponding features,
4. Optimal estimation of the model,
5. Estimate final orientation using lidar points.

The first three steps are carried out using the 2D EO images of the scene. The final 3D transformation to register the texel images is computed using the 3D points obtained

by the lidar sensor. Since the texel images are a fusion of texture information and range information, if the point clouds are registered, the texel images are also registered, and vice versa. The steps listed above are discussed more in detail in the following sections.

3.1.1 Detect Harris Features

The first step in matching the point clouds is to detect common features in the two EO images. A corner detector algorithm is used to find features, or distinct points, in the EO images. These features are of high importance since the set of features which are common to both the images are used to find the transformation between the two images. The corner detector algorithm was developed by Harris and Stephens [27]. This algorithm was an improvement on Moravec's corner detector [28].

The corner detector functions by considering a local window in the 2D image. The average changes in the image intensity are measured that result from shifting the window. The shifting is done such that all neighboring pixels are considered. The difference in the intensities for each shift are measured and compared to a threshold value, described in the Appendix, and the following inferences are drawn based on the changes noted.

1. If the intensity is approximately constant, i.e., the difference in the intensity is low over all the neighboring pixels, the window is flat, i.e., has no features.

2. If the window is at an edge, then the shift along the direction of the edge will result in small change, but a shift in the direction perpendicular to the edge will result in a large shift in intensity.

3. If the window is at a corner, then a shift in any direction will result in a large change in intensity. A corner can thus be detected when the least change in intensity due to any shifts is large.

Harris and Stephens improved upon Moravec's corner detector by making changes like using a circular window instead of a square one and made it more efficient by making the response close to isotropic, by shifting in all possible directions around the window. The

window size of the Harris corner detector is discussed in the Appendix. The basic properties of the corner finder algorithm is illustrated in Fig. 3.1.

The mathematical details are given in Harris' and Stephens' work [27]. Since the EO images captured by the texel camera are colored, they are divided into three image planes (R, G, B) and the corner finder algorithm finds feature points in all the image planes individually and combines them to form a list of feature points in the EO image. Figure 3.2 illustrates the feature detection with an example. The features that are detected using the Harris detector are marked with red points in the images.

3.1.2 Find Putative Correspondences

Since the goal is to find points that are common to both the images, the task after finding features in individual images is to find which feature in one image corresponds to a feature in the other image. This steps acts as a limiting filter for the next step, which involves finding a 2D transformation for the two EO images. The process of finding putative correspondences is based on correlation. The process is given as follows:

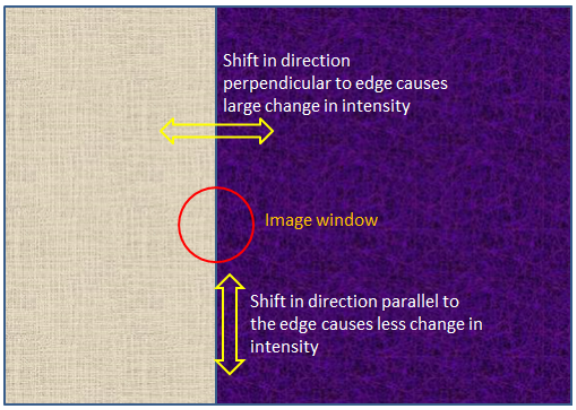
1. Measure the correlation score of a window centered about each feature in one image, with another window centered about each feature in the second image using (3.1) and (3.2).

2. If the correlation score is higher than a threshold, then the corresponding features are assumed to be putative correspondences.

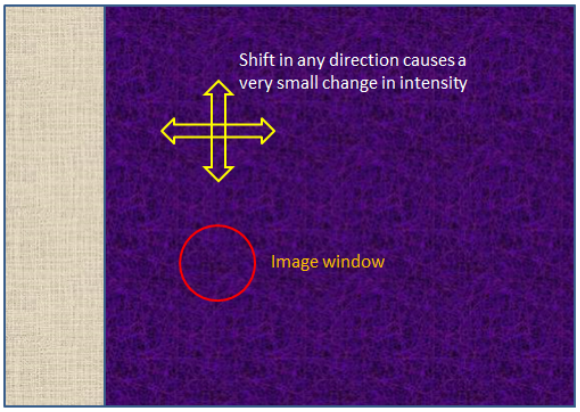
3. If multiple features in the second image have correlation scores with a feature point in the first image that are higher than the threshold, the feature in the second image with the highest correlation score is chosen as the corresponding feature point.

The correlation score (Xc) of a feature point in image 1 with a feature point in image 2 is calculated using:

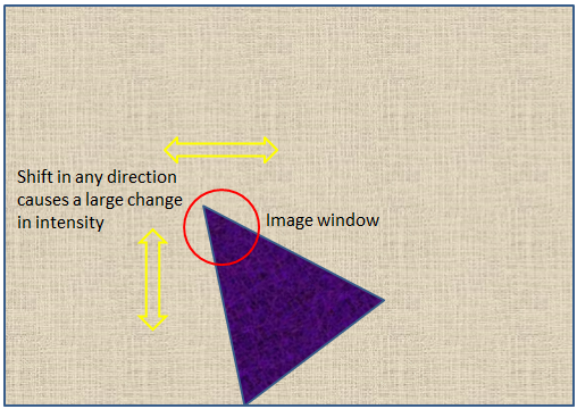
$$Xc = X_R + X_G + X_B, \tag{3.1}$$



(a) Harris feature detector window traversing an edge.

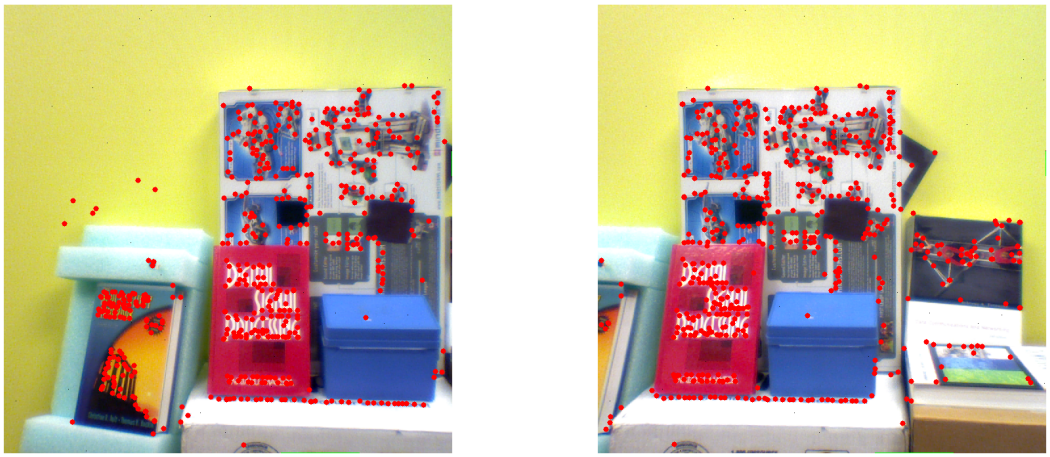


(b) Harris feature detector window traversing a plane surface.



(c) Harris feature detector window traversing a corner.

Fig. 3.1: Harris corner detection algorithm.



(a) Image 1. (b) Image 2.

Fig. 3.2: Harris features: results of the Harris corner detector for two EO images.

where

$$X_R = \sum_{r=-\frac{N_R}{2}}^{\frac{N_R}{2}} \sum_{c=-\frac{N_C}{2}}^{\frac{N_C}{2}} \frac{(I1_R(x+r,y+c)-\mu(I1_R))(I2_R(x+r,y+c)-\mu(I2_R))}{\sqrt{(I1_R(x+r,y+c)-\mu(I1_R))^2+(I2_R(x+r,y+c)-\mu(I2_R))^2}}$$

$$X_G = \sum_{r=-\frac{N_R}{2}}^{\frac{N_R}{2}} \sum_{c=-\frac{N_C}{2}}^{\frac{N_C}{2}} \frac{(I1_G(x+r,y+c)-\mu(I1_G))(I2_G(x+r,y+c)-\mu(I2_G))}{\sqrt{(I1_G(x+r,y+c)-\mu(I1_G))^2+(I2_G(x+r,y+c)-\mu(I2_G))^2}}$$

$$X_B = \sum_{r=-\frac{N_R}{2}}^{\frac{N_R}{2}} \sum_{c=-\frac{N_C}{2}}^{\frac{N_C}{2}} \frac{(I1_B(x+r,y+c)-\mu(I1_B))(I2_B(x+r,y+c)-\mu(I2_B))}{\sqrt{(I1_B(x+r,y+c)-\mu(I1_B))^2+(I2_B(x+r,y+c)-\mu(I2_B))^2}}. \tag{3.2}$$

In (3.2), N_R and N_C are the number of rows and columns in the correlation window, given in pixel coordinates. $I1_R(x,y)$ and $I2_R(x,y)$ are the intensities of the point at (x,y) in the first and second image, respectively, in R (red) image plane, where x and y is the row and column for the point, given in pixel coordinates. The average intensities in image 1 and image 2, computed over the correlation window centered about the feature points in

R image planes, are given by $\mu(I1_R)$ and $\mu(I2_R)$. Similar notations are used for G (green) and B (blue) image planes. The correlation window size and the correlation threshold value are described in the Appendix.

The results of the correlation based method are illustrated in Fig. 3.3. The correspondences found are marked with similar color points.

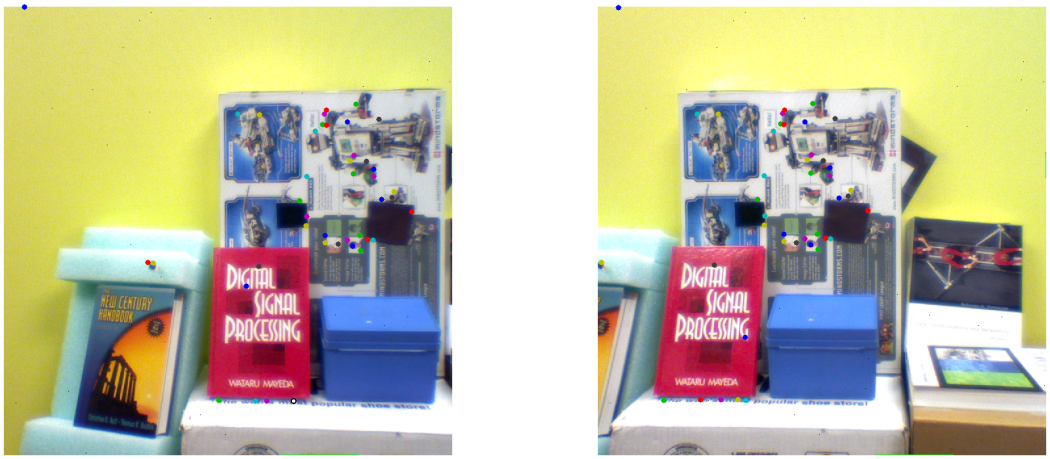
3.1.3 Establish a Model to Fit Putative Correspondences

Having a good correlation between features is not a sufficient condition to decide if they are matching points in the two images. There can be multiple features in an image that are similar to features in the other image. In such a case, finding correspondences based on the constraint of correlation only can result in incorrect matches. Hence, in order to find a model that fits most of the correct correspondences, called the inliers, and to eliminate the the wrong correspondences, the outliers, a parameter estimation technique called Random Sample Consensus (RANSAC) is used [29]. RANSAC does not depend on the assumption that most of the dataset is not corrupted with gross errors. RANSAC has the ability to fit the data set better and eliminate the outliers in the data sets. RANSAC generates a model, based on randomly picked data points and then decides the accuracy of the model based on how many how many of the total number of data points fit the model. The model used by Boldt with RANSAC to find putative correspondences is called the homography model [6].

Homography, also known as projective transformation, studies the effects of relations between linear spaces where any given point in open space corresponds to one and only point in any other linear space [30]. This model can be used to relate between images of the same scene, taken from different poses. For 2D images, this relationship can be defined as $\boldsymbol{x_A} = H\boldsymbol{x_B}$, where $\boldsymbol{x_A}$ and $\boldsymbol{x_B}$ are points in homogeneous coordinate system in the two images and H is a matrix having dimensions of (9×9). To determine a homography model, a minimum of four data points are required [30].

Using the model of homography, RANSAC is used to find the best model that fits the given set of putative correspondences as given in Algorithm 1.

Even though the model determined by the method described in Algorithm 1 fits many



(a) Image 1. (b) Image 2.

Fig. 3.3: Putative correspondences.

of the points in the data set, this model is not the optimal homography transformation, since it is computed using just four points out of the set of data points. The optimal transformation is computed using all the inliers obtained for the best model, instead of just

---

Algorithm 1 RANSAC to find the best model that fits the data points that satisfy the correlation constraint.

Input:

- List of putative correspondences, $(\boldsymbol{p}_i \leftrightarrow \boldsymbol{p}'_i)$
- Number of iterations, m

Output:

- Best fit homography model, H
- List of inlier points that fit H, $(P_t \leftrightarrow P'_t)$, t = 1..N

Begin

For m **Iterations**

- Randomly select four pairs from the putative correspondences
- Determine planar projection (homography) [30], using the correspondences chosen in the previous step.
- **For** All the putative correspondences
  - Count the pairs of correspondences that fit the homography model. Let this number be n.
- **End For**

- **End For**
- Choose the model with maximum n as the best fit model, H.

End

---

four used in RANSAC. This is calculated using Singular Value Decomposition [25].

3.1.4 Estimate Final Orientation Using Ladar Points

RANSAC gives the best 2D model that fits the two EO images. The final aim is to find a transformation such that the texel images are registered. Hence, a 3D transformation needs to be computed to match the individual point clouds. The following process uses the information available from previous processing to compute the 3D transformation.

1. Estimate the position of 2D feature points in terms of azimuth and elevation using the position on 2D image using a pixel to polar coordinate mapping.

2. Estimate range of the corresponding points by bilinear interpolation of the closest lidar points.

3. Compute the position of the point in 3D space in Cartesian Coordinates.

4. Use these points to determine 3D transformation for two images captured.

The 3D transformation, also know as the rigid-body transformation can be defined as $[\boldsymbol{R}, \boldsymbol{T}]$, where $\boldsymbol{R}$ is a 3 x 3 rotation matrix and $\boldsymbol{T}$ is a 3D translation vector. The transformation maps two corresponding point sets m_i and n_i, where $i = 1..\text{N}$, N being the number of point correspondences, such that they are related by

$$m_i = \boldsymbol{R} n_i + \boldsymbol{T} + noise. \tag{3.3}$$

Solving for an over-determined system, the optimal transformation $[\tilde{\boldsymbol{R}}, \tilde{\boldsymbol{T}}]$ is estimated by minimizing least squares error E^2 given by

$$E^2 = \sum_{i=1}^{N} \left(m_i - \tilde{\boldsymbol{R}} n_i - \tilde{\boldsymbol{T}}\right)^2. \tag{3.4}$$

The method to find a least squares solution, given an over-determined system is given by Arun *et al.* [31].

3.2 Improvements on the Existing Technique

As stated previously, this thesis proposes a method that improves upon the method described by Boldt and Nelson. A few changes were implemented in the process of fusing range information to texture information using a different mapping function described in section 2.4.3.4. The texel images obtained after the fusion were calibrated for errors in range using a method based on a LUT and flat-field correction as described in section 2.4.4. The calibration process reduces the error introduced due to the walk error in the 3D lidar points. Apart from these, a few iterations were added to the process of determining the final 3D transformation. The changes are listed as follows:

1. Added additional constraint to RANSAC using concepts of epipolar geometry.

2. Used planar interpolation instead of bilinear interpolation for better accuracy in finding 3D point coordinates from 2D image points.

3. Added a second iteration to the computation of the 3D transformation to eliminate "bad" points which were not eliminated by RANSAC or correlation in the previous iteration.

4. Added a third iteration, based on previous iterations, to add more points to the set of inliers that are used to determine the 3D transformation.

5. Computed the final 3D transformation by minimizing the reprojection error, using a nonlinear optimization method known as the Levenberg-Marquardt method.

The changes were incorporated in only a few steps of the method described by Boldt and Nelson. The rest of the steps, which included detection of Harris features, finding putative correspondences and the method of determining 3D transformation using a given set of corresponding 3D points, were unchanged. The changes made and the improvements caused by them are described in brief in the following sections.

3.2.1 RANSAC Based on Epipolar Geometry

The method described earlier computes a homography model using four randomly picked pairs of points and uses that model as a constraint to find inliers for that model. This method is made more general by using a model and a constraint based on epipolar geometry [16]. This method uses eight point correspondences to determine the model [32], and the transformation model based on the principle of epipolar geometry is known as the fundamental matrix [16].

Epipolar geometry is the geometry of stereo vision. As shown in Fig. 3.4, it is the geometry associated with the intersection of the individual image planes (image 1 and image 2) with the plane which is formed by the baseline (line joining the two camera centers) and a point on the object in 3D space, as given by the triangle formed by vertices C, C' and $\boldsymbol{X}$ in Fig. 3.4. Epipolar geometry gives a relationship between a point $\boldsymbol{X}$ in 3D space that is visible in two images captured from different perspectives, and the corresponding image points x and x', which are the projections of $\boldsymbol{X}$ on the two image planes. Figure 3.4 shows that the points x and x', 3D point $\boldsymbol{X}$ and camera centers C and C' are coplanar. This plane is known as the epipolar plane corresponding to point $\boldsymbol{X}$ and two camera centers, C and C'. The rays back-projected from x and x' intersect at $\boldsymbol{X}$, and lie on the epipolar plane of $\boldsymbol{X}$. The intersection of the epipolar plane with the image plane of C' is a line , say l'. This line is the image of the ray back-projected from x, on the image plane of C' and the point x' lies on this line . Thus, l' is the epipolar line corresponding to point x. The point x', which is the point corresponding to $\boldsymbol{X}$ in image plane of C', will be limited to the line l'. This constraint on the projection of a 3D point in the other image plane is called the epipolar constraint. Mathematically, it is given as

$$x'^{T}\boldsymbol{F}x = 0, \tag{3.5}$$

where F is the fundamental matrix [16] and x and x' are the homogeneous image coordinates.

The fundamental matrix is computed using a minimum of eight correspondences, as given by Hartley [32]. The best model that fits most of the points in the set of putative

correspondences is determined using the epipolar constraint in the RANSAC algorithm. This method is described in the work given by Budge and Badamikar [26]. It is given in Algorithm 2.

Once the best transformation model and the list of inliers that fit the model are obtained, the optimal transformation is computed using all the inliers obtained from RANSAC. This transformation is then used to test the list of putative correspondences to get a final list of inliers, resulting in a set of points that satisfy the correlation and the epipolar constraints. The results of RANSAC and the final test is given in Fig. 3.5, where the corresponding inliers that meet the correlation constraint and the epipolar constraint are marked by points of same color. The white lines are the epipolar lines corresponding to each inlier.

3.2.2 2D to 3D Transformation Using Lidar-to-Image Mapping and Planar Interpolation

The 3D points corresponding to the 2D inliers obtained after RANSAC are used to determine the 3D transformation. These 2D points do not lie exactly on the existing lidar pixels and hence, the 3D coordinates of the points of interest need to be calculated using interpolation between the existing 3D lidar points.

A set of parameters, $(\boldsymbol{u}, \boldsymbol{v})$, are used to map the 2D image coordinates to 3D lidar points in a texel image. The $(\boldsymbol{u}, \boldsymbol{v})$ give the 2D image coordinate to lidar point mapping where u corresponds to the value along the columns of the image (along x-axis) and the v value corresponds to the value along the rows (along y-axis). They are normalized, such that $0 \leq u, v \leq 1$, where $u = 0$ maps 2D image coordinates to 3D lidar point that is to the extreme left in 3D space, and $u = 1$ gives the mapping to the 3D point to the extreme right. Similarly, $v = 0$ corresponds to the top-most lidar pixel (along y-axis) and $v = 1$ corresponds to the bottom-most lidar pixel. The origin is assumed to be at the top-left corner of the 2D image. The $(\boldsymbol{u}, \boldsymbol{v})$ values are assigned such that the span along x-axis is equal to the span along y-axis, in order to maintain the square shape of pixels. Hence, for example, if the span along $y - axis$ for an image is more than the x-axis, then the $(\boldsymbol{u}, \boldsymbol{v})$ values are assigned in a way that the u value will range from 0 to 1 $(0 \leq u \leq 1)$, but the v

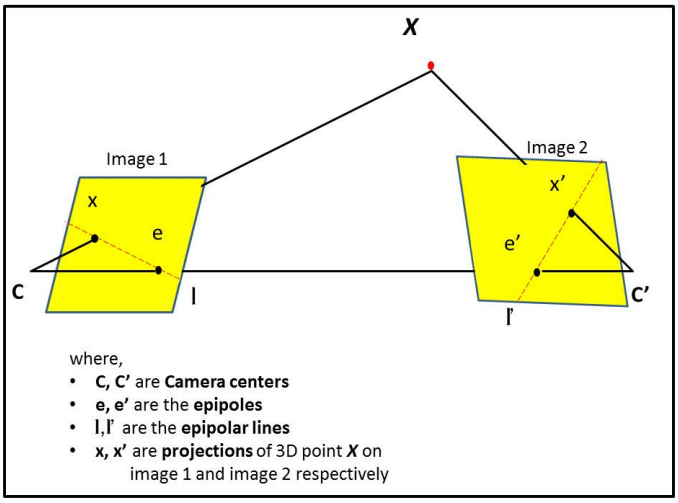


Fig. 3.4: Epipolar geometry.

Algorithm 2 RANSAC to find the best model based on epipolar constraint, that fits the data points that satisfy the correlation constraint.

Input:

- List of putative correspondences, $(\boldsymbol{p}_i \leftrightarrow \boldsymbol{p}'_i)$
- Number of iterations, m

Output:

- Best fit model (Fundamental matrix), F
- List of inlier points that fit F, $(P_t \leftrightarrow P'_t)$, t = 1..N

Begin

For m **Iterations**

- Randomly select eight pairs from the putative correspondences
- Determine fundamental matrix, F, as given in [32] using the correspondences chosen in the previous step.
- **For** All the putative correspondences
  - Check for inliers using epipolar constraint (3.5). Count the pairs of correspondences that satisfy (3.5) within a certain tolerance threshold. Let this number be n.
- **End For**

• **End For**

• Choose the model with maximum n as the best fit model, F.

End

value will range between 0 to a value n $(0 \leq u \leq n)$ that is linearly scaled using the ratio

of the span along x-axis and y-axis, given as

$$n = \frac{Span_x}{Span_y}, \tag{3.6}$$

where $Span_x$ is determined by computing the distance between the lidar point on extreme left and the lidar point on extreme right along x-axis, and $Span_y$ is determined by computing the distance between the top-most lidar point and the bottom-most lidar point along y-axis, using the 3D coordinates of the point.

Thus, the $(\boldsymbol{u}, \boldsymbol{v})$ mapping is such that the position of each lidar point corresponding to the 2D image coordinate is known. Thus any 2D point $\boldsymbol{T} = (p, q)$ can be converted to $(\boldsymbol{u}, \boldsymbol{v})$ values using

$$\begin{bmatrix} u_T \\ v_T \end{bmatrix} = \begin{bmatrix} \frac{p}{NumCol} \\ \frac{q}{NumRow} \end{bmatrix}, \tag{3.7}$$

where p, q are 2D coordinates of a point in pixel coordinates, and $NumRow$ and $NumCol$ are the row count and column count, respectively, in the 2D image.

The $(\boldsymbol{u}, \boldsymbol{v})$ mapping is used to determine the four lidar points in terms of lidar rows and columns that are closest to the 2D feature point. Since three points are necessary and

(a) Image 1. (b) Image 2.

Fig. 3.5: RANSAC using epipolar constraint.

sufficient to define a plane, only three closest lidar points out of the four points surrounding the 2D feature point are chosen and planar interpolation is used to obtain the 3D coordinates of the feature point.

Since the indices of the closest pixels are known in terms of row number and column number, their normalized values x_n^i and y_n^i, where $i = 1$, 2 or 3, can be used to compute the normalized values $\tilde{x}_n$ and $\tilde{y}_n$ of the point of interest.

The equation of a plane passing through three points in Cartesian Coordinate system is given by

$$\begin{vmatrix} (x-x_1) & (y-y_1) & (z-z_1) \\ (x_2-x_1) & (y_2-y_1) & (z_2-z_1) \\ (x_3-x_1) & (y_3-y_1) & (z_3-z_1) \end{vmatrix} = 0 = \begin{vmatrix} (x-x_1) & (y-y_1) & (z-z_1) \\ a & b & c \\ d & e & f \end{vmatrix}, \tag{3.8}$$

where (x_1,y_1,z_1), (x_2,y_2,z_2) and (x_3,y_3,z_3) are 3D coordinates of the measured lidar pixels. In (3.8), let

$$\begin{aligned} a &= (x_2 - x_1), \qquad b = (y_2 - y_1), \qquad c = (z_2 - z_1); \\ d &= (x_3 - x_1), \qquad e = (y_3 - y_1), \qquad f = (z_3 - z_1). \end{aligned} \tag{3.9}$$

The 3D coordinates of the point of interest, $\tilde{\boldsymbol{P}}$= ($\tilde{x}$,$\tilde{y}$,$\tilde{z}$), can be expressed using the normalized values in (2.1) as

$$\tilde{\boldsymbol{P}}_t = (\tilde{x}_n\tilde{z}, \tilde{y}_n\tilde{z}, \tilde{z}). \tag{3.10}$$

The 3D coordinates of the point of interest can be computed by substituting (3.10) in (3.8) and as the x, y, and z point in the plane equation and solving for $\tilde{z}$,

$$\tilde{z} = \frac{x_1(bf - ce) - y_1(af - dc) + z_1(ae - bd)}{-\tilde{x}_n(bf - ce) - \tilde{y}_n(af - dc) + (ae - bd)}. \tag{3.11}$$

The actual 3D coordinates for the point of interest can then be computed using (3.10).

3.2.3 Eliminating Points on Edges

It is observed that 3D lidar points that lie on edges of surfaces which have a large difference in depth (along z-axis) are more prone to error or inaccurate measurement. This is because the surface is oblique to the transmitter and the receiver of the lidar sensor. The reflectivity of the surface is affected due to the angle, and hence the information received is prone to noise.

An effort is made to check and select only those points which were on flat or relatively plain surfaces in the texel images for determining the 3D rigid body transformation. A two-fold test is conducted on each 2D point feature.

1. Determine the three lidar points that are closest to the feature point using planar interpolation, as described in section 3.2.2. Calculate the difference in depth between all the three points, taking one pair at a time. If the maximum difference in the points is less than a threshold distance, pick the point as a valid inlier. Else proceed to step 2.

2. Use the three lidar points determined in step 1 to compute the plane equation and the equation of the normal to the plane. Determine the angle made by the normal and the ray incident on the plane from the origin, i.e., COP of the sensor. If the angle is less than a threshold value of the angle (described in the Appendix), pick the point as a valid inlier.

The above mentioned test ensures that the 3D point picked as an inlier does not lie on an edge. This is ensured by determining the angle made by the normal to the plane and the incident ray, and checking if it is less than a certain threshold angle. One possibility, however, exists such that the feature point lies on a flat surface, which is parallel or close to parallel to the sensor array, yet the angle made by the plane in which it lies has an angle greater than the threshold value set. Since the range measurements of the lidar sensor are noisy, the difference in depths due to noise can cause the angle of the plane with the sensor array to be larger than the threshold value, if the three points that define the plane are closer to each other in x and y directions. Since the points are closer, the small difference in

depth causes the plane to tilt, resulting in an angle which may sometimes be larger than the threshold angle used to eliminate points on edges. In order to prevent such false elimination of points, step 1 checks if the maximum difference in depths is large enough to be certain of the point lying on an edge, and only then proceed to step 2.

If the angle made by the normal and the incident ray (step 2) is large, the plane on which the 3D point lies is expected to be at an angle, larger than the threshold value, with the sensor array of the lidar sensor. This could mean that the plane lies along a fringe or an edge, and so does the point lying on the plane. This possibility is tested in step 2 of the given test.

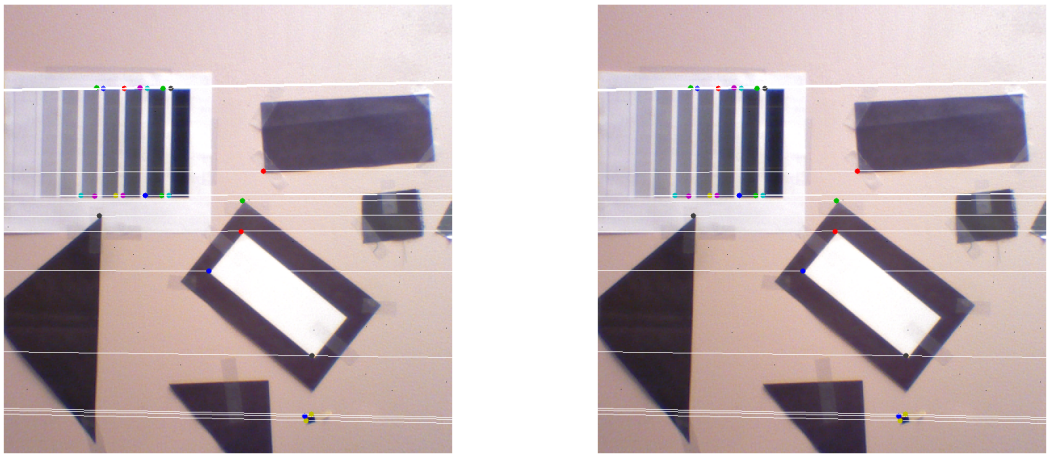
If a point fails both the tests, it can be concluded that the point lies on an edge and is discarded.

3.2.4 A Corrective Iteration to Improve Accuracy of Inlier Points

It can be seen that the epipolar constraint and the correlation constraint are necessary, but not sufficient conditions for a pair points to be correct inliers. Any pair of points which are highly correlated, and lie on the epipolar line can be assumed to match correctly, even though they actually may not be the same point in 3D space. This is illustrated in Fig. 3.6. The points marked with dots having same color are the points which are picked as inliers. It can be seen that some points are picked as inliers, inspite of being incorrect correspondences. As shown in Fig. 3.6, the points are highly correlated and satisfy the epipolar constraints since they lie on the same epipolar line as the correct correspondences do.

A solution to this is to use the 3D transformation estimated in the first iteration to guide the selection of matching points from the list of putative correspondences. Assuming that the first transformation is reasonably close to being correct, the initial $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{T}}$ can be used to eliminate incorrect correspondences.

The epipolar test given in section 3.2.1 results in a set of correspondences that may be one-to-many for each feature point in the first EO image. This is likely to happen when multiple points in the second EO image satisfy both the epipolar constraint and



(a) Image 1. (b) Image 2.

Fig. 3.6: Incorrect matches inspite of meeting correlation constraint and the epipolar constraint.

the correlation threshold for a single point in the first EO image. The one-to-many list of correspondences needs to be pruned such that the correct feature point is picked, or alternately, the bad correspondence is discarded from the list of inliers.

Choosing of incorrect correspondences as inliers can result from two cases. One, if the one-to-many list contains the correct correspondence for a given feature point, yet the wrong one is chosen because of a higher correlation score with the feature point. The second case is when the correct correspondence does not exist in the one-to-many list. This might happen if a 3D point is picked as a corner by the Harris corner detector in the first image, but not in the second image. As a result, some other point, which has the correlation score higher than the threshold and satisfies the epipolar constraint may be chosen as a likely correspondence. Both the cases result in incorrect choice of correspondences, hence leading to a poor solution.

A correcting method is used, either to rectify the incorrect choice of correspondences, which can be possible in the first case mentioned above, or discard the incorrect correspondence completely, which works for the second case.

Given the list of 2D inliers in first EO image, $\boldsymbol{x}$, and the initial estimate of the 3D

transformation, $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{T}}$, each 3D point corresponding to the 2D point x_i in first EO image is transformed into 3D coordinate system of second image using $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{T}}$ and then projected into the second EO image as 2D point, $\tilde{x'}_i$. If the feature point in the first image and its corresponding feature point in the second image is a correct match, the projected point $\tilde{x}_i$ should be such that the distance between $\tilde{x'}_i$ and the corresponding feature point x'_i, in second EO image, is very small. By using this principle, a two-step corrective test is conducted in order to ensure that the correspondence for a given feature point is correct.

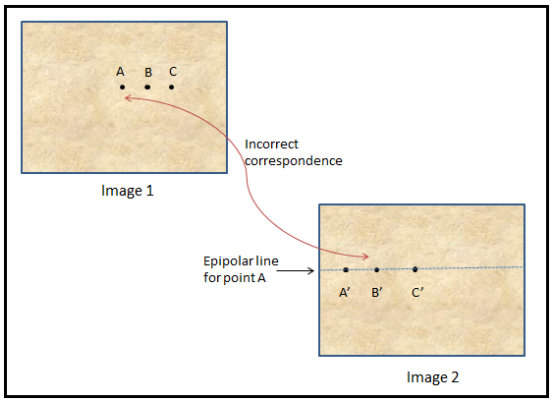
1. From the one-to-many list of correspondences in the second image, chosen for the point x_i in the first image, the feature point x_i having the smallest Euclidean distance to the projected point $\tilde{x'}_i$ is selected.

2. If the Euclidean distance between the selected point and the projected point $\tilde{x'}_i$ is below a threshold value (described in the Appendix), which is chosen to be a small value, the selected point is picked as the correspondence in second image for point x_i in the first image.

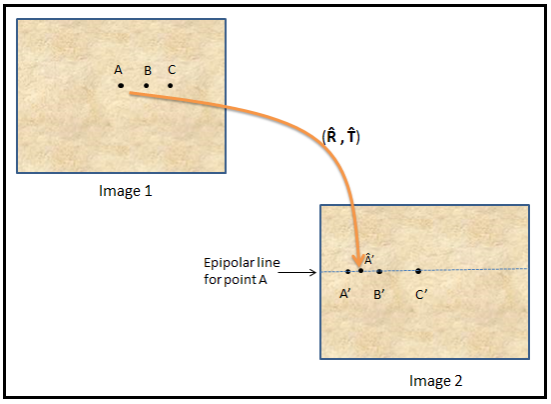
If none of the points in the many-to-one list of correspondence pass the corrective test, the point x_i and its correspondence is removed from the list of inliers. The process of correcting an incorrect match or discarding the match altogether is illustrated in Fig. 3.7 and Fig. 3.8.

Figure 3.7 illustrates the case of correcting an incorrect match, given that the correct match exists in the one-to-many list of correspondences for a given feature point in the first image. The points are incorrectly matched if multiple points satisfy the epipolar constraint and an incorrect match point, satisfying the epipolar constraint, has a higher correlation score than the correct point. This case in illustrated in Fig. 3.7(a). The points A', B' and C' satisfy the epipolar constraint and have correlation scores with point A which are higher than the threshold. Hence, the one-to-many list of correspondences for point A consists of points A', B' and C'.

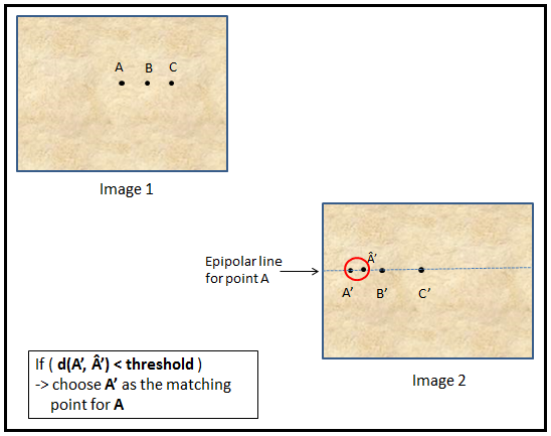
Even though points $A \leftrightarrow A'$ are correct pair of inliers, the pair of $A \leftrightarrow B'$ is chosen to be the correct one. As mentioned before, this can be the result of points B' and A having a



(a) Step 1.



(b) Step 2.



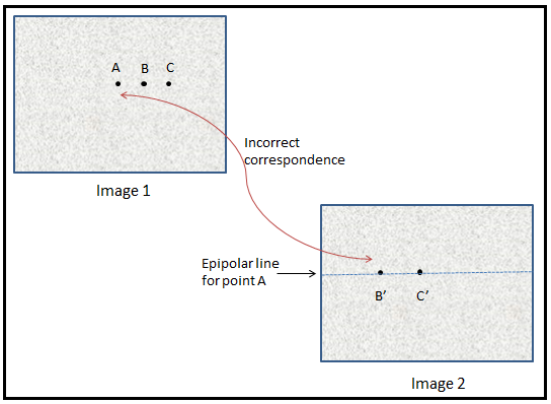
(c) Step 3.

Fig. 3.7: Corrective iteration: correcting inaccurate correspondences.

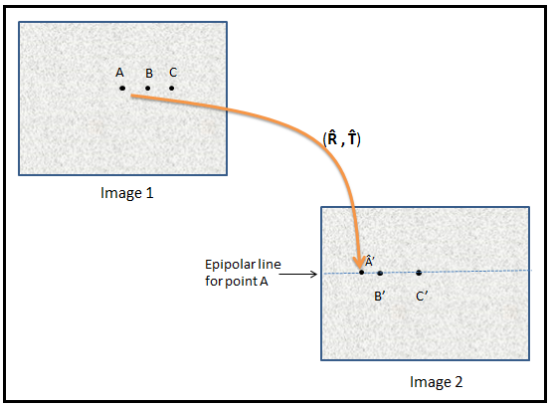
higher correlation score than the correlation score of points A and A'. It can be seen that both the points A' and B' satisfy the epipolar constraint. Since the correct correspondence for point A exists, this pair of inliers can be corrected. Using $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{T}}$, the 3D point in the first lidar image, corresponding to the point A in the first EO image, is transformed into the 3D coordinate system of the second texel image. This point is then projected as a 2D point $\tilde{A}'$ on the second EO image, as shown in Fig. 3.7(b). Since the 3D transformation $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{T}}$ is close to correct, if not accurate, the point $\tilde{A}'$ is projected close to its correct match, A', and the Euclidean distance between points $\tilde{A}'$ and A' is the least in the one-to-many list of inliers for the point A. Hence, according to the corrective test, point A' is chosen as the correct match for point A, as shown in Fig. 3.7(c).

Figure 3.8 illustrates the case of eliminating an incorrect match by discarding the point correspondence altogether. An incorrectly matched pair of inliers is formed if the correct match for the feature point in first image does not exist as a feature point in the second image. Instead, a correspondence is picked from a list of points which satisfy the epipolar constraint and correlate with the feature point in the first image. This case in illustrated in Fig. 3.8(a). The points B' and C' satisfy the epipolar constraint and have correlation scores with point A which are higher than the threshold. Hence, the one-to-many list of correspondences for point A consists of points B' and C'.

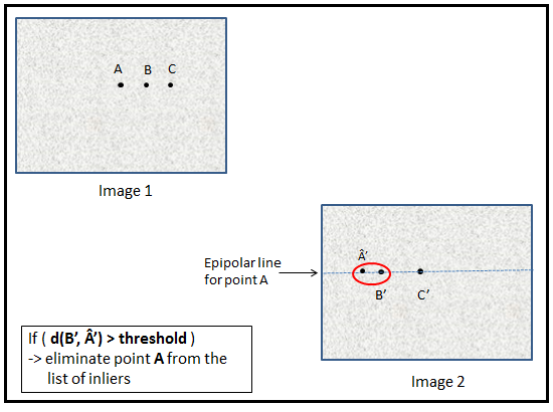
Even though the correct match for point A does not exist in the list of feature points in the second image, the pair of $A \leftrightarrow B'$ is chosen to be the correct match. Using $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{T}}$, the 3D point in the first lidar image, corresponding to the point A in the first EO image, is transformed into the 3D coordinate system of the second texel image. This point is then projected as a 2D point $\tilde{A}'$ on the second EO image, as shown in Fig. 3.8(b). Since the 3D transformation $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{T}}$ is close to correct, the point $\tilde{A}'$ is projected such that the Euclidean distance between point $\tilde{A}'$ and B' is greater than the threshold value set. Hence, according to the corrective test, since no point in the one-to-many list satisfy the conditions in the test, the correspondence is discarded and point A and its matching correspondence is removed from the list of inliers, as shown in Fig. 3.8(c). Since the correct correspondence



(a) Step 1.



(b) Step 2.



(c) Step 3.

Fig. 3.8: Corrective iteration: discarding inaccurate correspondences.

for point A does not exist, this pair of inliers cannot be corrected.

All the inliers are checked and only those which are correct or are corrected are retained. The 3D transformation is recomputed using the method described in section 3.1.4 using the retained inlier points as inputs. The resulting 3D transformation results in a new $\tilde{\boldsymbol{R}}_2$ and $\tilde{\boldsymbol{T}}_2$, where the subscript indicates the iteration in which the solution is determined.

3.2.5 Recomputing 3D Transformation Using Additional Points

The number of points used to determine the transformation is usually larger than the minimum number of points required, which is eight [32], resulting in an over-determined system. A least squares solution is estimated using the points selected [31]. For a least squares solution of an over-determined system, more data points usually result in a better least-squares fit. Hence, to get additional points which are corresponding inliers in the two images, an additional iteration is introduced in the algorithm.

For the least squares fitting method used to compute the 3D transformation between two texel images, the best solution would be obtained if all the points that lie in the region of overlap of the two texel images are in the list of inliers, without any incorrect matches. That is difficult to achieve, since existing techniques enable the user to identify distinct points only, like Harris features, and use them as inliers with good confidence. Using similar logic, a large set of inliers which is distributed uniformly over the region of overlap gives a better result than a set of points concentrated in a small area of the overlapping region. Hence, this step aims at generating more points in the texel images that can be assumed to be matching correctly, in order to get a better 3D transformation.

These additional points are selected from the list of Harris feature points detected previously. They are selected using the transformation computed in the previous iteration, which is assumed to be close to correct. This ensures that the added points are not outliers but instead are corresponding feature points within tolerance of the initial 3D transformation. The additional points generated in this iteration are used along with those used to determine the previous transformation to compute the final 3D transformation ($\tilde{\boldsymbol{R}}_3$, $\tilde{\boldsymbol{T}}_3$) using the method described in section 3.1.4.

The method proposed to add new points uses the list of Harris features in both the images, the 3D transformation ($\tilde{\boldsymbol{R}}_2$, $\tilde{\boldsymbol{T}}_2$) computed using method suggested in section 3.2.4 and the test used to eliminate points which lie on edges of 3D objects in the scene. The steps involved in selecting new points is given in Algorithm 3.

3.2.6 Optimizing the 3D Transformation Using Nonlinear Optimization

The 3D transformation is estimated using the method described in section 3.1.4 is a linear solution. Using this 3D transformation as an initial guess, a better transformation is recomputed by minimizing the reprojection error, which is calculated using the 3D points in the two image planes and the transformation that maps points in one image onto the other. This method involves adjusting the values of individual elements that form the 3D

Algorithm 3 Addition of new Inlier points to compute the 3D transformation.

Input:

- Count of Harris Features in first and second image, N1 and N2.
- List of Harris feature points detected in the first image, ($\boldsymbol{x}_i$), i=1...N1,
- List of Harris feature points detected in the second image, ($\boldsymbol{x'}_k$), k=1...N2,
- 3D transformation ($\tilde{\boldsymbol{R}}_2$, $\tilde{\boldsymbol{T}}_2$)

Output:

- List of additional 3D inlier points, ($P_t \leftrightarrow P'_t$), t = 1..M

Begin

- **For** Each Harris feature point (x_i) in first image
  - **If** 3D point X_i, corresponding to x_i, does not lie on an edge in the 3D scene
    - Using ($\tilde{\boldsymbol{R}}_2$, $\tilde{\boldsymbol{T}}_2$), transform $X_i \rightarrow \tilde{X}'_i$.
    - **If** $\tilde{X}'_i$ is in the FOV covered by second image
      - Project $\tilde{X}'_i$ on 2D EO image from the second texel image to get $\tilde{x}'_i$.
      - **For** Each Harris feature point (x'_k) in second image
        - By computing Euclidean distance, d($\tilde{x}'_i$, x'_k), find the feature point x'_k closest to $\tilde{x}'_i$.
        - **If** The d($\tilde{x}'_i$, x'_k) $\leq$ **distance threshold**,
          - Find 3D point X'_k corresponding to x'_k.
          - Store X_i and X'_k as P_t and P'_t.
        - **Else** Discard x_i and choose next feature point.
      - **End For**
    - **Else** Discard x_i and choose next feature point.
  - **Else** Discard x_i and choose next feature point.
- **End For**

End

---

transformation, as well as the 3D coordinates of the points used to determine the transformation to account for noisy measurements of the lidar sensor.

3.2.6.1 Reprojection Error

Reprojection error is an error based on geometric distance between measured and estimated coordinates in the respective images. Unlike other distance-based error functions, reprojection error takes into account the error in measured points in both image coordinates. In the present case, we seek a 3D transformation ($\tilde{\boldsymbol{R}}$, $\tilde{\boldsymbol{T}}$) and pairs of 3D inlier points ($\tilde{\boldsymbol{x}}_{\boldsymbol{i}} \leftrightarrow \tilde{\boldsymbol{x}}'_{\boldsymbol{i}}$) that minimize the total error function given by

$$E_{rep}^2 = \sum_{i=1}^{N} d(x_i - \tilde{x}_i)^2 + d(x'_i - \tilde{x}'_i)^2, \tag{3.12}$$

where

$$\tilde{x}_i = \tilde{\boldsymbol{R}} x'_i + \tilde{\boldsymbol{T}}. \tag{3.13}$$

Points x_i and x'_i in (3.12) are the 3D points measured by the lidar sensor. These measured points are assumed to be noisy and a noise-free estimate of these points, $\tilde{x}_i$ and $\tilde{x}'_i$, respectively, is obtained by minimizing the reprojection error, along with correcting the 3D transformation. The concept of reprojection error is explained in more detail by Hartley and Zisserman [16].

3.2.6.2 Levenberg Marquardt Method for Nonlinear Optimization

A method called the Levenberq Marquardt algorithm is implemented to minimize the reprojection error in order to improve upon the linear solution obtained by the method described in section 3.1.4. The minimization proceeds iteratively. Given trial values for parameters, the procedure attempts to improve the trial solution. This procedure is then repeated till the error reduces below a given threshold value. This method closely resembles

the gradient descent method to find a local minima for a known error function. This method is described in brief by Press *et al.* [33].

The function to be minimized is given as an error function ($\chi^2(\boldsymbol{a})$), where $\boldsymbol{a}$ is the list of m parameters which are to be adjusted in order to minimize the error function. In this case, the error to be minimized is the reprojection error given by (3.12) and (3.13). This procedure develops iteratively and each iteration takes the solution one step closer to the optimum solution. This step is denoted by $\boldsymbol{\delta}$, which is a vector with m elements. The step $\boldsymbol{\delta}$ gives a small correction value for the parameters to be optimized ($\boldsymbol{a}$), in order to reduce the error defined by $\chi^2(\boldsymbol{a})$, such that $\chi^2(\boldsymbol{a}) > \chi^2(\boldsymbol{a} + \boldsymbol{\delta})$. The step $\boldsymbol{\delta}$ taken towards the optimum solution depends on the first derivative; i.e., the gradient, a m element vector $\boldsymbol{\beta}$, and the second derivative; i.e., the Hessian, a $m \times m$ square matrix $\boldsymbol{\alpha}$, of the error function $\chi^2(\boldsymbol{a})$. The step in each iteration is computed by solving for the following equation

$$\sum_{l=1}^{N} \boldsymbol{\alpha}_{k,l} \boldsymbol{\delta}_{a_l} = \boldsymbol{\beta}_k, \tag{3.14}$$

where

$$\boldsymbol{\beta}_k = \frac{\partial \boldsymbol{\chi^2}}{\partial a_k}, \tag{3.15}$$

and

$$\boldsymbol{\alpha}_{kl} = \frac{\partial^2 \boldsymbol{\chi^2}}{\partial a_k \partial a_l}. \tag{3.16}$$

Equation (3.14) corresponds to the steepest descent formula and translates to

$$\boldsymbol{\delta_l} = k \cdot \boldsymbol{\beta_l}. \tag{3.17}$$

The constant k is given by $\lambda \boldsymbol{\alpha}_{ll}$, where λ was introduced as a damping factor to scale the step size. The damping factor is used to scale the diagonal elements of $\boldsymbol{\alpha}$. Thus step $\boldsymbol{\delta}$ can be computed using steepest descent formula as

$$\boldsymbol{\delta_l} = \frac{1}{\lambda \boldsymbol{\alpha}_{ll}} \boldsymbol{\beta_l}. \tag{3.18}$$

Incorporating (3.18) in (3.14) to get a new set of equations

$$\sum_{l=1}^{N} \boldsymbol{\alpha}'_{k,l} \boldsymbol{\delta}_{a_l} = \boldsymbol{\beta}_k, \tag{3.19}$$

where

$$\begin{aligned} \boldsymbol{\alpha}'_{jj} &= \boldsymbol{\alpha}_{jj}(1+\lambda), \\ \boldsymbol{\alpha}'_{jk} &= \boldsymbol{\alpha}_{jk}. \end{aligned} \tag{3.20}$$

This equation can be solved linearly to compute $\boldsymbol{\delta}$.

The Levenberg Marquardt method can be summarized in the following steps:

1. Compute $\chi^2(\boldsymbol{a})$,
2. Pick a small value for λ, such as λ= 0.01,
3. Solve linear equation (3.19) for $\boldsymbol{\delta}$ and evaluate $\chi^2(\boldsymbol{a}+\boldsymbol{\delta})$,
4. If $\chi^2(\boldsymbol{a}+\boldsymbol{\delta}) \geq \chi^2(\boldsymbol{a})$, increase λ by a factor of 10 and go back to step 3,
5. If $\chi^2(\boldsymbol{a}) \geq \chi^2(\boldsymbol{a}+\boldsymbol{\delta})$, decrease λ by a factor of 10 and update trial solution $\boldsymbol{a} = \boldsymbol{a} + \boldsymbol{\delta}$.

Thus, using the Levenberg Marquardt method, the 3D transformation ($\tilde{\boldsymbol{R}}$, $\tilde{\boldsymbol{T}}$) is optimized after each iteration. The linear solution ($\tilde{\boldsymbol{R}}_n$, $\tilde{\boldsymbol{T}}_n$) is used as a starting point for this method.

Chapter 4

Results

The Camera Calibration methods described in Chapter 2 and the image matching techniques given in Chapter 3 were implemented and then tested experimentally on different setups. These setups consisted of different arrangements of plain colored boards, checkered boards, specific patterns that would help observe a particular effect and various 3D objects. The results obtained on these experiments are discussed in this chapter. The first part of this chapter presents the results of various steps of the calibration process and later the results of the point cloud matching algorithm are discussed.

4.1 Calibration Results

The calibration as given in Chapter 2 consisted of three major steps:

1. Correcting the lidar image for lens distortions,
2. Mapping the EO image on the lidar image, while correcting the EO image for lens distortions,
3. Correcting the lidar image for range errors.

The results for each of these steps are discussed. These results are also given in the article presented by Budge and Badamikar [12].

The lidar camera was calibrated, and the lens distortion parameters given in (2.3)-(2.5) and the camera matrix given in (2.7) were computed using the CCTM [14]. These parameters were used to estimate normalized image coordinates $(\tilde{x}_n, \tilde{y}_n)$ for all the detectors on the sensor using the method described in section 2.4.3.2. These values are plotted in Fig. 4.1 as a grid, representing the pixel positions of the lidar sensor. It can be seen that the lidar pixels are not uniformly positioned with respect to each other. The array is not

rectangular due to the lens distortion. This type of distortion, where the outer pixels in the array are pulled towards the center, is known as pin-cushion distortion. The values of different parameters obtained from the CCTM are listed in Table 4.1.

The lidar to image mapping was computed as described in section 2.4.3.4. The mapping parameters can be estimated using (2.16). Since this mapping uses lidar points that are calibrated so that they are linear, the mapping parameters correct for the lens distortions of the EO sensor as well. This eliminates lens distortions in the EO image. This is shown in Fig. 4.2. It can be seen that the raw image is barrel distorted, and the mapping corrects for the distortion, which results in the corrected image being almost linear.

The mapping of the EO image on the lidar pixels is shown in Fig. 2.4. It can be seen

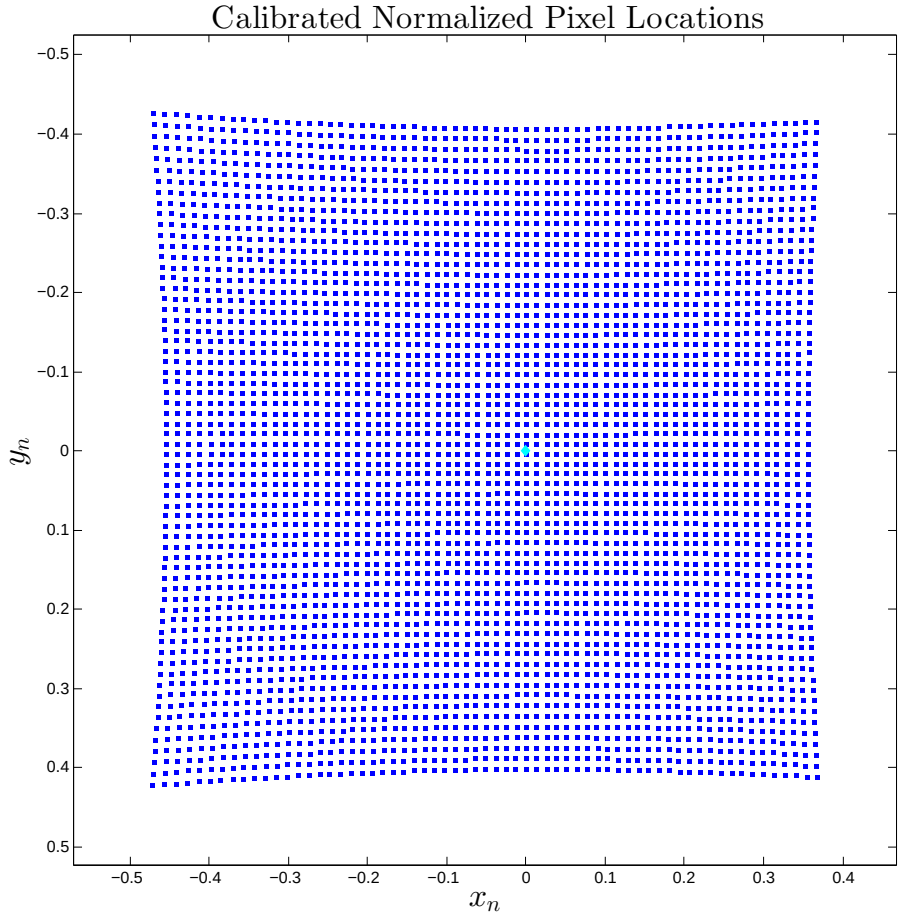


Fig. 4.1: Corrected lidar pixels: the position of lidar pixels to remove lens distortions.

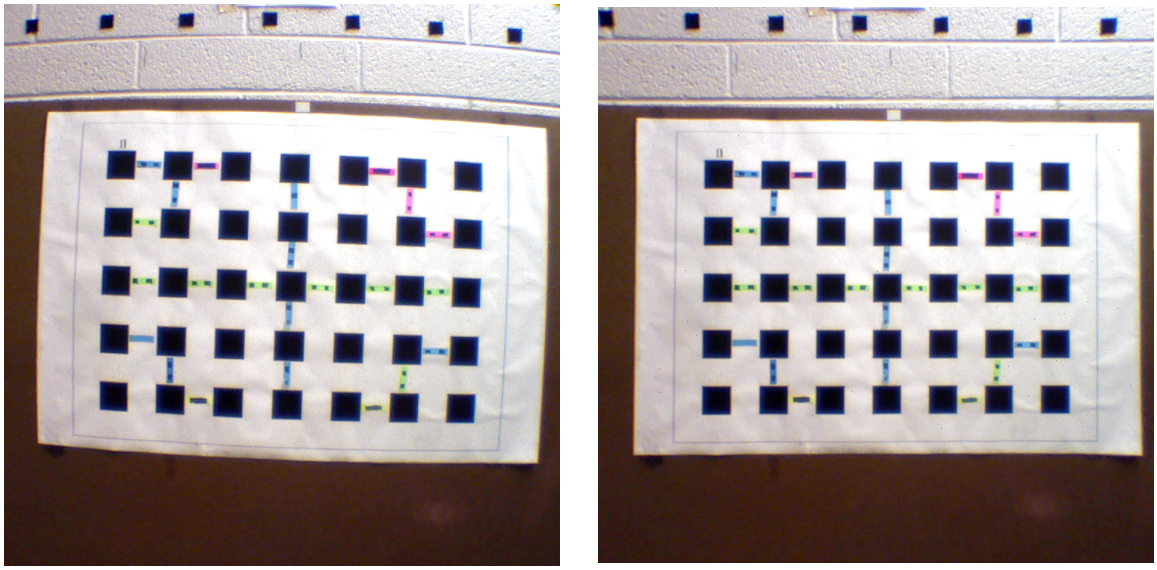
Table 4.1: Calibration results: distortion parameters for the lidar sensor.

Parameter	Symbol	Value	Tolerence
Focal Distance	(f_x,f_y)	(80.4527 , 80.3708)	± (0.2318 , 0.24581)
Principal Point	(c_x,c_y)	(34.8945, 31.6703)	± (0.3637, 0.3564)
Skew	(s)	0.0000	0.0000
Distortion Parameters:	(k_1)	-0.19969	± 0.01315
	(k_2)	0.05126	± 0.04457
	(k_3)	-0.00077	± 0.0.00079
	(k_4)	0.00411	± 0.00083
	(k_5)	0	± 0.0000
Pixel Error (σ_x,σ_y)	(0.06693, 0.06836)		

that the individual images captured by the lidar sensor and the EO sensor match very well. The edges and corners of the 3D objects match in both the images, showing that the FOVs of both the sensors match correctly.

The texel images were corrected for range error using the method described in section 2.4.4. The results were tested using a target which consisted of eight bars of different shades of a dark color, each representing a different reflectance for IR light. This target was chosen such that the effect of walk error due to difference in the amount of IR light reflected by different colors was observed. The error was observed to be reduced, but was not completely eliminated. This is illustrated in the Fig. 4.3. Figure 4.3(c) represents the raw texel image without any range calibration. It can be seen that walk error is evident on pixels that correspond to the bars that have dark shades. The effect of walk error can be seen more significantly in Fig. 4.3(e) which shows a colored representation in terms of range for the same texel image. Figure 4.3(d) and Fig. 4.3(f) represent the texel image that has been corrected for range using the calibration LUT. It can be seen that the walk error in the corrected image is less as compared to the raw image.

The final step involved testing the accuracy of points projected in the 3D space. This was carried out by comparing the distances between known points in the texel image computed using the 3D coordinates of the points of interest and the actual distances between the corresponding points in the real world. The distance computed using the 3D points is denoted as “measured distance” and the true world distance is denoted as “true distance.” This experiment was carried out for different positions and distances of the target from



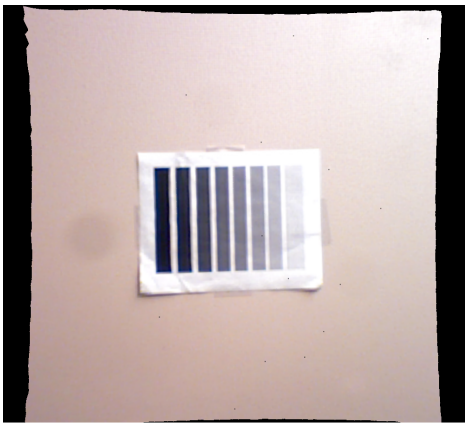
(a) Raw image. (b) Corrected image.

Fig. 4.2: Correction for lens distortion for the EO sensor.

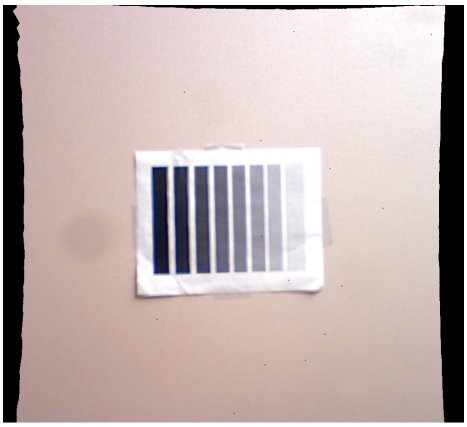
the texel camera. The error associated with each distance, mean error and the standard deviation for each experiment were recorded. The results are in illustrated in Table 4.2.

The first two columns show mean error and the standard deviation in the measured distances between known points in the texel images obtained for different depths, without any corrections for the lens distortion or range error. The following two columns show the results after incorporating the corrections for the lens distortions given in (2.3)-(2.5). This shows that the error is reduced by about 30% by incorporating the nonlinear correction methods. The next two columns illustrate the results after implementing correction for the nonlinearities and includes an estimated value of the z_o offset between the COP of the lidar sensor and the origin assumed by the sensor firmware. It can be seen that the after correcting for z_o, the error reduces to about 3% of the initial mean error in the measured distances. The z_o computed has a value of 2.98 cm (towards the focal plane).

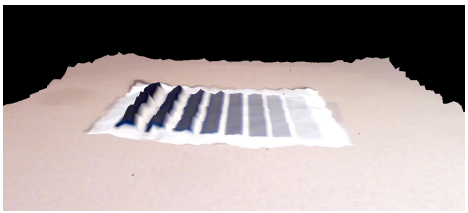
The final two columns show the results after all the calibration steps are completed, including the flat-field correction and the correction for wiggling error and walk error using the LUT, as described in section 2.4.4.1 and section 2.4.4.2. It was observed that the average flat-field correction offset for all the lidar pixels was 2.0 cm with a variance of 0.23 cm. This



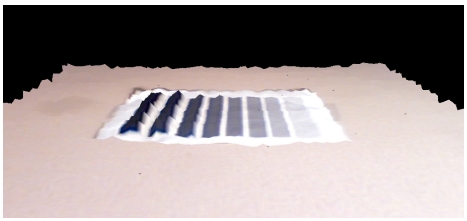
(a) Front view of the raw texel image.



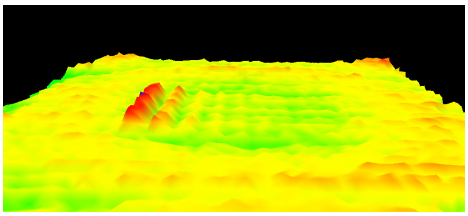
(b) Front view of the corrected texel image.



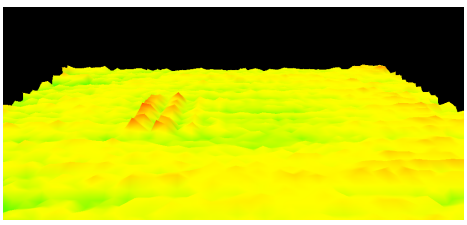
(c) Top view of the raw texel image.



(d) Top view of the corrected texel image.



(e) Top view of the raw texel image with colormap showing the error in depth.



(f) Top view of the Corrected texel image with colormap showing the error in depth.

Fig. 4.3: Range calibration: the correction in the distortion due walk error is seen in the figures showing the corrected texel image.

offset was included before the calculation of z_o. The z_o was recomputed after the correction for wiggling error, walk error, and the flat field correction, and was found to be 0.87 mm. This shows that the reference point assumed during the flat field correction is close to the actual COP of the lidar sensor. It can be seen from the results presented in Table 4.2 that the mean error and the standard deviation for the measured distances reduce after each step of calibration is implemented. The final error is observed to be only 1.26% of the original error.

4.2 Point Cloud Matching Results

The calibration process improved the accuracy of the texel images, which were used in the Point Cloud matching algorithm given in Chapter 3. The algorithm was implemented and tested using various flat and 3D targets.

The Harris corner detector, described in section 3.1.1, was used to find distinct corners or feature points which were used to compute the transformation later. The algorithm was tested using different 2D images. Figure 4.4 illustrates the result of the corner detector for set of images of a bookshelf setup that were captured from different perspectives.

The list of features in both the images was sorted to find feature points that are common to both the images using the correlation technique, as mentioned in section 3.1.2. Though not sufficient, this condition is good enough to eliminate majority of mismatches. The results of the correlation based method to find the putative correspondences is illustrated in Fig. 4.5. The corresponding pairs of matching points are marked with dots of same color

Table 4.2: Calibration results: point cloud measurement error after each calibration step.

Depth cm. (z_r)	$z_o = 0$				$z_o = 2.98$ cm		$z_o^{opt} = 0.87$ mm	
	$k_1 ... k_5 = 0$		$k_1 ... k_5$ from calibration values		$k_1 ... k_5$ from calibration values		$k_1 ... k_5$ from calibration values	
	No Range Calibration		No Range Calibration		No Range Calibration		Range Calibrated	
	μ	σ	μ	σ	μ	σ	μ	σ
50	2.222	1.540	1.715	0.845	0.287	0.276	0.026	0.198
70	1.398	0.567	1.067	0.406	0.013	0.276	0.143	0.270
90	1.294	0.523	0.797	0.420	-0.293	0.371	-0.113	0.276
110	1.414	0.604	0.884	0.535	-0.180	0.356	0.025	0.245
Overall	1.582	0.796	1.116	0.671	-0.043	0.385	0.020	0.260

in the images taken of the bookshelf setup.

RANSAC was used to compute a best-fit model (fundamental matrix) from the set of putative correspondences. Each pair of putative correspondences is subject the epipolar constraint described in section 3.2.1, and the best model is determined by counting the number of inliers for each model. The points that satisfy the epipolar constraint and are chosen as inliers are shown in Fig. 4.6. The matching pair of points in the two images are marked with the same color and the white lines are the epipolar lines corresponding to each feature.

Once the 2D correspondences are found, they were projected into 3D space using the method described in section 3.2.2 and the 3D transformation $(\tilde{\boldsymbol{R}}_1, \tilde{\boldsymbol{T}}_1)$ was determined using a least-squares method, as described in section 3.1.4. The 3D transformation computed is optimized further by adjusting the values of the 3D points used to compute it and the transformation itself. This is accomplished using the Levenberg Marquardt method, described in section 3.2.5. It was observed that this further reduced the reprojection error which was computed using all the points used to determine the transformation. The graph of reprojection error in meter vs. iteration count for one particular instance is shown in Fig. 4.7.

The initial transform $(\tilde{\boldsymbol{R}}_1, \tilde{\boldsymbol{T}}_1)$, computed using the 3D point correspondences that were determined using just the correlation and the epipolar constraint, was observed to be close to correct, though not exactly correct for all target setups. It was seen that though most of the inliers selected using RANSAC were correct, in some cases some of the points did not match correctly. The conditions that would result in such erroneous matching and the method to improve the accuracy of the matches or discard them altogether using a corrective iteration are discussed in section 3.2.4. The transformation $(\tilde{\boldsymbol{R}}_2, \tilde{\boldsymbol{T}}_2)$ is recomputed in the corrective iteration. The Levenberg Marquardt method is used to optimize the 3D transformation $(\tilde{\boldsymbol{R}}_2, \tilde{\boldsymbol{T}}_2)$ after it is computed using linear least squares fitting method. It uses $(\tilde{\boldsymbol{R}}_2, \tilde{\boldsymbol{T}}_2)$ and the list of inlier points used to compute the linear solution as the starting point for the algorithm.

(a) Image 1. (b) Image 2.

Fig. 4.4: Harris features: results of the Harris corner detector for two EO images.

(a) Image 1. (b) Image 2.

Fig. 4.5: Putative correspondences.

The advantage of the corrective iteration is evident in Fig. 4.8, which shows how the corrective iteration helps in improving the registration of two texel images of a flat wall setup having some geometrical figures and patterns of different colors embedded on it. Figure 4.8(c) shows the registration of texel images based on transformation computed using incorrectly matching points. The inaccurate transformation $(\tilde{\boldsymbol{R}}_1, \tilde{\boldsymbol{T}}_1)$ results in mismatch

(a) Image 1. (b) Image 2.

Fig. 4.6: RANSAC based on epipolar geometry: inlier pairs marked with same colors.

in the registration as seen in regions close to the grey bar chart and the broken edges of the black square and the rectangle on the right side of the merged texel image. The transformation $(\tilde{\boldsymbol{R}}_2, \tilde{\boldsymbol{T}}_2)$ was observed to improve significantly after the corrective iteration was implemented and Fig. 4.8(d) shows evident improvement in the registration of the texel images. The mismatches seen in Fig 4.8(c) are largely corrected.

The quantitative results of the corrective iteration are shown in Table 4.3. The MSE is observed to reduce significantly after the transformation is recomputed using correct pairs of inlier points.

The transformation $(\tilde{\boldsymbol{R}}_2, \tilde{\boldsymbol{T}}_2)$ was then used to compute additional points from the list of features. The computation of additional points helps in finding more common feature points in the two images in addition to the ones which have already been found. A larger number of feature points give more data to compute the 3D transform. A better fit is obtained if the feature points are well distributed over the region of overlap, than having a points concentrated in a smaller area. The main purpose of this step is to add more points to the set from which the 3D transform is computed, without the addition of significant error to the transform. The steps involved in the process of adding more points are described in section 3.2.5. The added set of inlier points are appended to the list of points used to

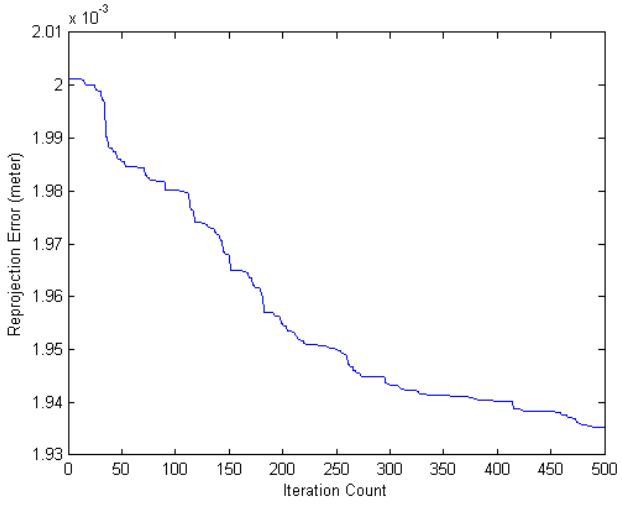


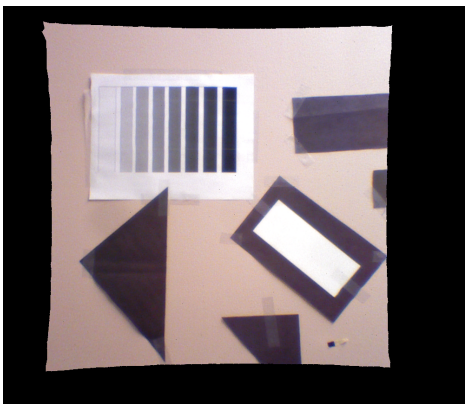
Fig. 4.7: Reprojection error vs iteration count: graph showing the result of the Levenberg Marquardt method.

determine the transformation $(\tilde{\boldsymbol{R}}_2, \tilde{\boldsymbol{T}}_2)$. The 3D transformation is recomputed by the least squares method, as described in section 3.1.4, using the extended list of inlier points, and is denoted as $(\tilde{\boldsymbol{R}}_3, \tilde{\boldsymbol{T}}_3)$. The transformation $(\tilde{\boldsymbol{R}}_3, \tilde{\boldsymbol{T}}_3)$ is then optimized further using the Levenber Marquardt method.

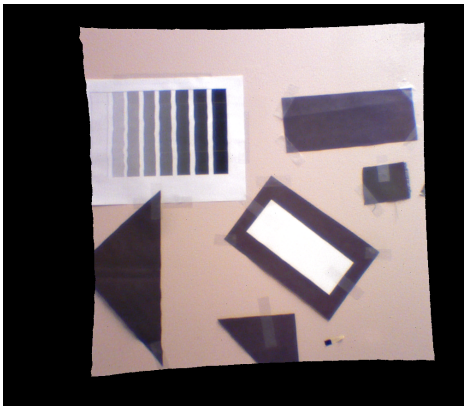
Table 4.4 shows the results of the third iteration for the three different target setups used as examples in this thesis. It can be seen that the MSE increases for two out of the three target setups. The reason for this is the addition of extra points, and hence extra noise, to the existing set of points. The noise results from the measurement errors introduced due to the techniques used to compute these points, like the Harris corner detector or the interpolation method used to project a 3D point on the EO image or to back-project a 2D point into 3D space. However, it can be seen that the MSE remains somewhat constant after the additional points are added.

The results for matching the texel images of the 3D setup and a bookshelf are illustrated in Fig. 4.9 and Fig. 4.10, respectively.

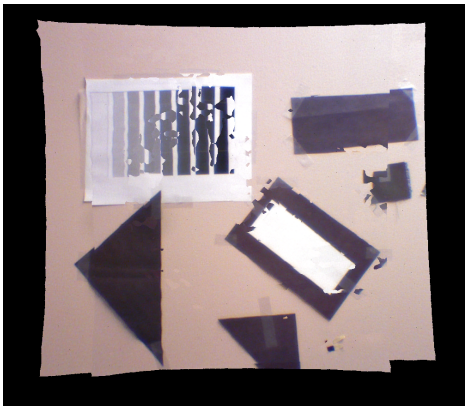
Thus, it can be seen that the methods described in Chapter 3 work satisfactorily and the matching is observed to improve with each additional iteration that was implemented.



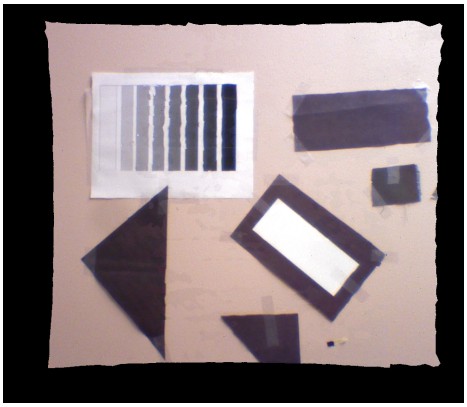
(a) Front view of the first texel image.



(b) Front view of the second texel image.



(c) Matching before the corrective iteration.



(d) Matching after the corrective iteration.

Fig. 4.8: Illustration of the improvement due to the corrective iteration.

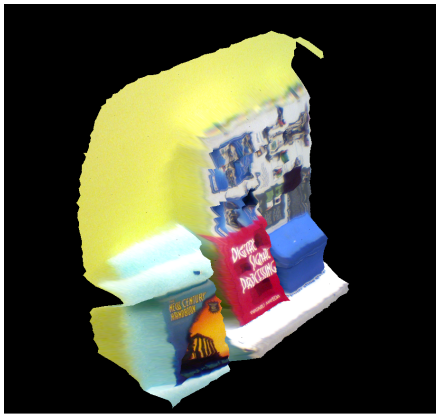
Table 4.3: Results: corrective iteration.

Setup	Transformation $(\tilde{\boldsymbol{R}}_1, \tilde{\boldsymbol{T}}_1)$		Transformation $(\tilde{\boldsymbol{R}}_2, \tilde{\boldsymbol{T}}_2)$	
Description	Point Count	MSE (Meter squared)	Point Count	MSE (Meter squared)
Flat Wall	20	0.00043	16	0.000073

Table 4.4: Results: after adding additional feature points.

Setup	Before Adding Additional Points		After Adding Additional Points	
Description	Point Count	MSE (Meter squared)	Point Count	MSE (Meter squared)
Flat Wall	16	0.000073	19	0.000069
3D Setup	64	0.000040	74	0.000063
Bookshelf	32	0.000057	40	0.000058

(a) Front view of the first texel image.



(b) Left top view of the first texel image.

(c) Front view of the second texel image.

(d) Left top view of the second texel image.

(e) Front view of the combined texel images.

(f) Left Top view of the combined texel images.

Fig. 4.9: Example of 3D registration implemented on texel images of a 3D setup.

(a) Front view of the first texel image.

(b) Left top view of the first texel image.

(c) Front view of the second texel image.

(d) Left top view of the second texel image.

(e) Front view of the combined texel images.

(f) Left top view of the combined texel images.

Fig. 4.10: Example of a matching implemented on a bookshelf setup.

Chapter 5

Conclusions and Future Work

The calibration methods described in Chapter 2 were tested and found to be effective in creating texel images which are significantly more immune to lens distortions in either sensors. The resulting texel images yield good results for the matching of lidar and EO images and the 3D objects in the image are represented with accurate dimensions, which is reflected in the results given in Chapter 4. The calibration process described in Chapter 2 reduces the average error in the 3D measurements in a texel image from 1.582 cm to 0.02 cm, which is 1.26 % of what the error is when the texel image is not calibrated.

Individual sensors capture images simultaneously. This reduces the possibility of error introduced due to the time difference between the events of image capturing by individual sensors. This advantage of simultaneous image capturing is seen significantly when the camera or the target is in motion, causing the view of the target to change with time. Also, both the sensors and the cold mirror are positioned such that the FOV is the same for both the sensors, reducing parallax between individual sensors.

The calibration procedure described in Chapter 2 uses parameters with fixed values. These values need not be computed during the capture process, but can be used directly from a LUT. As a result, the texel images are calibrated at low computing costs. This property, coupled with the advantage of simultaneous capture and fusion of point cloud and EO image at the time of acquisition makes use of texel images possible in real-time applications like automatic target recognition or security and surveying purposes. Though the calibration methods specified in this thesis were tested using a TOF lidar, they can be applicable to any camera that uses the combination of lidar sensor and EO sensor.

The images can be used to create 3D images of a scene by taking multiple images of the scene from different positions using the method described in this thesis. Results show

that the method performs well in registering 3D images to form a mosaic describing the scene with texture and range information. Since the texel image is a combination of both texture (2D) and point cloud (3D), the advantages of both the 2D and 3D registration techniques are combined to overcome the drawbacks in methods that use either of them individually. The problem associated with the scale ambiguity faced in decomposition of 2D transformation is avoided due to the fact that 3D points, with known range data, are used to determine the transformation. The requirement of preprocessing for the point clouds to provide information about the overlap that is necessary for ICP to work is unnecessary due to the use of 2D EO images to find inlier points in the region of overlap, as described in Chapter 3.

5.1 Drawbacks of the Point Matching Algorithm

The main drawback of the Point Cloud matching algorithm described in this thesis is that the process begins with extracting feature points in 2D images of the scene which works only if there is adequate light incident on the scene. This completely eliminates the possibility of using images taken in darkness and limits its usability. For example, if the camera is mounted on a UAV for surveillance purposes, this method will not work at times when the ambient light is not adequate or in areas where the light is not sufficient for the Harris corner detector to work efficiently, and the user will be "blinded."

Since the correlation windows in this method are fixed in terms of size and shape, the effect of difference in perspectives in the two images is observed when the disparity or the angle between the two positions from where the images are captured is large. This leads to incorrect correlation matches or inability to find correct putative correspondences.

Another drawback is that the added iterations after the initial computation of the 3D transformation depends on the initial transformation to be close to correct. If the initial transformation is far off from correct, due to incorrect matches, the second iteration would fail to correct the incorrect matches efficiently. Since each iteration depends on the transformation computed in the previous iteration and assumes it to be close to correct, this may lead to divergence of error with successive iterations and eventually cause the method

to fail.

5.2 Future Work

The work reported in this thesis is analogous to the tip of the iceberg and much more work can be done on this topic to improve the results, the efficiency and to include corrective measures in the algorithm.

This thesis gives a method to match the 3D points in the texel images. The 3D points are matched using the 3D transformation, merging the two point clouds. The texture, however, is fused with the individual point clouds and is merged according to the registration of point clouds. This poses a problem of the texture in one image overlapping that of the other image in the region of overlap when the two texel images are registered. Since texel images are 2.5D images, the texture is mapped on the point clouds such that the points that are not in the FOV are not visible to the sensor, and hence do not appear in the texel images. Texel image of a scene taken from one perspective contains visual and range information about the object that may not be measured by the sensors while capturing the texel image from the other perspective. Hence, when such images are registered, even though the point clouds are registered correctly, the overlapping texture information prevents viewing of the actual texture of the object. This overlap of texture information could be avoided by devising a technique to register texture information that can be fused on the registered point clouds.

Another drawback of the method described in this thesis is the dependence on 2D image processing technique to begin with. Alternative methods using the same logic of Harris corner detectors to detect features or corners on point clouds, instead of 2D images can be implemented and added to the algorithm given in this thesis as an alternate starting point in the case of failure of the Harris corner detector on 2D EO images. This will improve the reliability and robustness of this method, enabling it to work without relying on the ambient light while capturing images.

A method can be developed which will enable the algorithm to adapt the correlation window shape and size based on the angle or pose, in order to be able to detect the matching

features more efficiently. This will result in more points being included to the list of inliers and thus help the matching to improve.

Multiple threshold values are used in the point cloud matching algorithm, and the values are assumed to be known before the images are processed. These values, however, are not the same for all types of targets. A method can be developed to adjust these values based on the merits of the images. This would truly facilitate real-time usage of this method.

References

[1] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[2] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," *Third International Conference on 3-D Digital Imaging and Modeling. Proceedings.*, pp. 145–152, 2001.

[3] S.-Y. Park and M. Subbarao, "An accurate and fast point-to-plane registration technique," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 2967 – 2976, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865503001570

[4] J. Huang and S. You, "Point cloud matching based on 3d self - similarity," *Computer vision and Pattern Recognition Workshops*, pp. 41–48, 2012.

[5] F. Schouwenaars, R. Timofte, and L. Van Gool, "Robust scene stitching in large scale mobile mapping," *Proceedings BMVC 2013*, pp. 1–11, 2013.

[6] B. Boldt, "Point cloud matching with a Handheld texel camera," Master's thesis, Utah State University, Logan, UT, 2007.

[7] A. V. Jelalian, "Laser radar systems," *EASCON'80; Electronics and Aerospace Systems Conference*, vol. 1, pp. 546–554, 1980.

[8] A. V. Jelalian, W. H. Keene, C. M. Sonnenschein, C. E. Harris, and C. E. Morrow, "Fm-cw laser radar system," U.S. Patent 4,721,385, 1 26, 1988.

[9] R. T. Pack and F. B. Pack, "3d multispectral lidar," U.S. Patent 6,664,529, 12 16, 2003.

[10] R. E. Altenhofen and R. Hedden, "Transformation and rectification," *Manual of Photogrammetry Third Edition*, vol. 2, pp. 1–59, 1966.

[11] D. A. Kerr. (2005) The proper pivot point for panoramic photography. [Online]. Available: http://doug.kerr.home.att.net/pumpkin/Pivot_Point.pdf

[12] S. E. Budge and N. S. Badamikar, "Calibration method for texel images created from fused flash lidar and digital camera images," *Optical Engineering*, vol. 52, no. 10, p. 103101, 2013.

[13] J. Heikkila and O. Silven, "Calbration procedure for short focal legth off-the-shelf ccd camers," *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 1, pp. 166–170, 1996.

[14] J.-Y. Bouguet. (2007) Camera calibration toolbox for matlab. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[15] T. Melen, "Geometrical modelling and calibration of video cameras for underwater navigation," Ph.D. dissertation, Institutt for Teknisk Kybernetikk, Universitetet i Trondheim, Norges Tekniske Høgskole, 1994.

[16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.

[17] M. Frank, M. Plaue, H. Rapp, U. Köthe, B. Jähne, and F. A. Hamprecht, "Theoretical and experimental error analysis of continuous-wave time-of-flight range cameras," vol. 48, no. 1, p. 013602, Jan. 2009. [Online]. Available: http://link.aip.org/link/?JOE/48/013602/1

[18] M. Lindner and A. Kolb, "Calibration of the intensity-related distance error of the PMD TOF-camera," in D. P. Casasent, E. L. Hall, and J. Röning, Eds., vol. 6764, no. 1. SPIE, 2007, p. 67640W. [Online]. Available: http://link.aip.org/link/?PSI/6764/67640W/1

[19] T. Kahlmann, F. Remondino, and H. Ingensand, "Calibration for increased accuracy of the range imaging camera Swissranger," *Image Engineering and Vision Metrology (IEVM)*, pp. 136–141, 2006.

[20] S. T. Barnard and M. A. Fischler, "Stereo vision," *Encyclopedia of Artificial Intelligence*, pp. 1083 – 1090, 1987.

[21] G. Medoni and R. Nevatia, "Segment based stereo matching," *Computer Vision, Graphics, Image Processing*, vol. 31, pp. 2 – 18, 1985.

[22] T. Tuytelaars and L. van Gool, "Matching widely separated views based on affine invariant regions," *International Journal of Computer Vision*, vol. 59, no. 1, pp. 65–81, 2004.

[23] T. Kanade, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 920 – 932, 1994.

[24] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," *Pattern Recognition, 16th International Proceedings*, vol. 3, pp. 545 – 548, 2002.

[25] B. Nelson, "Image-based correction of ladar pointing estimates to improve merging of multiple ladar point clouds," Master's thesis, Utah State University, Logan, UT, 2006.

[26] S. E. Budge and N. Badamikar, "Automatic registration of multiple texel images (fused ladar/digital imagery) for 3d image creation," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2013, p. 873107.

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

[28] H. Moravec, "Obstacle avoidance and navigation in real world by a seeing robot rover," *Tech Report CMU-RI-TR3, Carnegie Mellon University, Robotics Institute*, 1980.

[29] M. Fischer and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Graphics and Image Processing*, vol. 24, pp. 381–395, 1981.

[30] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd ed. Florence: Thomas Learning, 2008.

[31] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698–700, 1987.

[32] R. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[33] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C*, 2nd ed. New York : Cambridge University Press, 1998.

Appendix

Explanation of the Parameters Used in Point Cloud Matching Algorithm

1. Threshold for Harris feature detection: the Harris corner detector measures the change in intensity between a window centered on one pixel, and the shifted window in all directions around the pixel. The threshold value decides the amount of the change in intensity between the window at a point and the shifted window around that point for the point to be selected as a feature. Lowering the threshold will increase the number of features detected.

2. Window size for Harris feature detection: this parameter decides the size of the window used in the Harris corner Detector. This parameter is defined in pixel coordinates and depends on the other parameters like the resolution and noise in the image. Having a larger window improves the performance against noise, but the corner to be detected needs to be large as well. A smaller window detects a smaller corner, but also is prone to noise and can detect incorrect corners due to intensity changes caused by noise or bad pixels.

3. Threshold for correlation: the threshold for correlation comes into the picture when deciding the putative correspondences. The higher the correlation threshold, the more accurate the putative correspondences are. However, increasing the correlation threshold results in loss of some putative correspondences which may be affected by noise.

4. Window size for correlation: the correlation window is a square window over which the correlation of two feature points is calculated. The size of the window depends on the noise in the image and also its resolution. A smaller window size is preferred for

accuracy, however one runs a risk of reducing the number of putative correspondences as the effect of difference in perspective is higher on smaller correlation windows.

5. Threshold for angle made by normal to plane: the threshold for angle made by the normal to the plane in which the 3D point corresponding to a 2D feature point is used to decide if a point lies on an edge or ridge. This threshold is set close to 45 degrees in order to accommodate for the noise in the lidar image.

6. Distance threshold for the corrective iteration: distance threshold is the maximum distance given in pixel coordinates that the closest feature point must lie within from the point projected from one image to the other. This threshold must be small enough to avoid incorrect points to be selected, but must be large enough to accommodate the noise in the images and the errors in the first transformation.