Statistical Methods for 2D-3D Registration of Optical and LIDAR Images

by

Dana Andrew Mastin

B.S., Electrical Engineering
University of Arizona, 2007

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2009

© Massachusetts Institute of Technology 2009. All rights reserved.

ARCHIVES

Author

Department of Electrical Engineering and Computer Science

September 4, 2009

Certified by

John W. Fisher III
Principal Research Scientist
Thesis Supervisor

Accepted by

Terry P. Orlando
Chair, Department Committee on Graduate Students
Statistical Methods for 2D-3D Registration of Optical and LIDAR Images

by

Dana Andrew Mastin

Submitted to the Department of Electrical Engineering and Computer Science on September 4, 2009, in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering and Computer Science

Abstract

Fusion of 3D laser radar (LIDAR) imagery and aerial optical imagery is an efficient method for constructing 3D virtual reality models. One difficult aspect of creating such models is registering the optical image with the LIDAR point cloud, which is a camera pose estimation problem. We propose a novel application of mutual information registration which exploits statistical dependencies in urban scenes, using variables such as LIDAR elevation, LIDAR probability of detection (pdet), and optical luminance. We employ the well known downhill simplex optimization to infer camera pose parameters. Utilization of OpenGL and graphics hardware in the optimization process yields registration times on the order of seconds. Using an initial registration comparable to GPS/INS accuracy, we demonstrate the utility of our algorithms with a collection of urban images.

Our analysis begins with three basic methods for measuring mutual information. We demonstrate the utility of the mutual information measures with a series of probing experiments and registration tests. We improve the basic algorithms with a novel application of foliage detection, where the use of only non-foliage points improves registration reliability significantly. Finally, we show how the use of an existing registered optical image can be used in conjunction with foliage detection to achieve even more reliable registration.

Thesis Supervisor: John W. Fisher III
Title: Principal Research Scientist
Acknowledgments

I would first like to thank Jeremy Kepner and groups 102, 104, and 106 at the Lincoln Laboratory for funding my first year of graduate studies, providing a stimulating research project, and helping with a variety of technical details. I am greatly indebted to my advisor John Fisher for general guidance, tolerating my erratic schedule, and convincing me to take the two inference courses during my first year of studies. I thank Alan Willsky for valuable discussions, as well as Fredo Durand and Kevin Chen for their help with the graphics-related parts of my work.

I acknowledge the NDSEG Fellowship Program, the Office of Naval Research, and the National Science Foundation for their generous financial support of my graduate studies.

The many friends I have met at MIT have greatly contributed to my graduate student experience. I thank Ryan Odegard for inviting me to live with him during my first year at MIT and always bombarding me with requests to go to Wendys. This last year would not have been the same without Paul Monasterio, who was always up for staying out late on weeknights or grabbing a bite to eat at the hotel. I believe the two of us single-handedly convinced the hotel that a 45% student discount on food was too high.

I am grateful for my officemates throughout the last two years, particularly Jason Chang and Archana Venkataraman. In addition to reading my thesis, Jason has proved to be a limitless source of creative schemes for saving money. Archana was always willing to help with questions related to 6.437 and qualifying exams, and was rarely without the correct answer. I thank Emily Fox and Michael Siracusa, as well as others in Vision, Graphics, and SSG for enjoyable discussions and valuable advice. I am also grateful for Raquel Santos and the many friends outside of MIT that I have met in the Boston area.

Most importantly, I thank my parents for their unwavering support. I owe much of my success to their advice and encouragement.
Contents

1 Introduction 17

2 Background 21
  2.1 Previous Work 21
  2.2 Airborne LIDAR 23
  2.3 Camera Model 25
  2.4 Ground Truth Registration 27
  2.5 Mutual Information Registration 29
  2.6 Quasiconvexity 31

3 Basic Image Registration 33
  3.1 Attribution Methods 34
    3.1.1 Elevation 34
    3.1.2 Pdet 40
    3.1.3 Dual 44
  3.2 Probing Experiments 45
  3.3 Registration Algorithm 53
  3.4 Registration Tests 54

4 Foliage Detection 67
  4.1 Detection Method 68
  4.2 Analysis 70
  4.3 Probing Experiments 76
List of Figures

2-1 Collection method for airborne LIDAR. The scene is scanned with a laser to measure range (shown on the left), producing a point cloud (shown on the right). .................................................. 24
2-2 Geometric diagram of camera model .................................................. 26
2-3 Optical image (a) and LIDAR image (b) with 30 manually selected correspondence points .................................................. 28

3-1 Optical image (a) and LIDAR image (b) of an urban scene. The different colors in the LIDAR image indicate different elevations. .................. 36
3-2 LIDAR elevation image (a) and optical luminance image (b) used to generate the PMFs in Figure 3-3. The blue regions in (a) indicate areas where no LIDAR points were projected into the 2D image plane. .................. 37
3-3 Log probability mass function (PMF) for the joint LIDAR elevation values and optical luminance values (a) and product of marginals (b). The mutual information calculated from the PMF is 0.3795 bits with a maximum MI of 5.2064 bits. .................................................. 38
3-4 Log probability mass function (PMF) of the LIDAR elevation values and optical luminance values (c) for the images shown in (a) and (b). The images are of the same scene, but have not been registered to illustrate the lower mutual information. The blue regions in (a) indicate areas where no LIDAR points were projected into the 2D image plane. The mutual information calculated from the PMF is 0.1431 bits with a maximum MI of 5.2064 bits. .................................................. 39
3-5 Optical image luminance (a) and pdet-encoded LIDAR image (b) of an urban scene.

3-6 LIDAR pdet image (a) and optical luminance image (b) used to generate the PMFs in Figure 3-7. The blue regions in (a) indicate areas where no LIDAR points were projected into the 2D image plane.

3-7 Log probability mass function (PMF) for the joint LIDAR pdet values and optical luminance values (a) and product of marginals (b). The mutual information calculated from the joint PMF is 0.2539 bits with a maximum of 7.473 bits.

3-8 Graphical model describing conditional independence relationship of LIDAR elevation, LIDAR pdet, and image luminance.

3-9 Images used for probing experiments and registration experiments.

3-10 Plots of normalized joint entropy for Image 1 and Image 2. Each camera parameter is smoothly varied while the other parameters are held constant at their ground truth values.

3-11 Plots of normalized joint entropy for Image 3 and Image 4. Each camera parameter is smoothly varied while the other parameters are held constant at their ground truth values.

3-12 Plots of normalized joint entropy for Image 5 and Image 6. Each camera parameter is smoothly varied while the other parameters are held constant at their ground truth values.

3-13 Plots of normalized joint entropy for Image 7 and Image 8. Each camera parameter is smoothly varied while the other parameters are held constant at their ground truth values.

3-14 Fade between LIDAR image and optical image with an $\alpha$ perturbation of one degree.

3-15 Histogram of mean squared error for initial perturbations (a) and post-algorithm registrations (b). The initial perturbations were drawn from the large parameter range. The histograms were generated with samples for all eight test images.
3-16 Example of discrepancies in image modes due to different collection times. The registration algorithm achieves the correct registration (c) despite the difference. ........................................ 58

3-17 Bar chart for values given in Table 3.3. The error bars indicate the minimum and maximum values. ........................................ 59

3-18 Bar chart for values given in Table 3.4. The error bars indicate standard deviations. ........................................ 60

3-19 Bar chart for values given in Table 3.6. The error bars indicate the minimum and maximum values. ........................................ 62

3-20 Bar chart for values given in Table 3.7. The error bars indicate standard deviations. ........................................ 63

3-21 Example of registration results with image superimposed on projected LIDAR image. The perturbation was sampled from the small range parameters. ........................................ 65

3-22 Texture mapped scenes made from post-algorithm registrations .... 66

4-1 Example of foliage detection performance. Detected points are shown in green while the non-foliage points are show with elevation-encoded grayscale values. Blue pixels indicate background regions. .......... 70

4-2 Example of foliage detection performance. Detected points are shown in green while the non-foliage points are show with elevation-encoded grayscale values. Blue pixels indicate background regions. The square shows a region with false alarms due to building edges; the circle shows false alarms resulting from vehicles. ........................................ 71

4-3 Foliage detection results for different parameters of $r$ and $\sigma^2$. The units are meters and meters$^2$, respectively. The LIDAR elevation and pdet images are shown for comparison. ........................................ 72

4-4 Optical image (a) and LIDAR image (b) showing which points were classified as foliage for the PMFs in Figure 4-5 and Figure 4-6. ...... 73
4-5 Joint elevation-luminance PMF and product of marginals PMF for non-foliage regions. The image used for calculating the PMFs and the classification of points is shown in Figure 4-4. ................. 74

4-6 Joint elevation-luminance PMF and product of marginals PMF for foliage regions. The image used for calculating the PMFs and the classification of points is shown in Figure 4-4. ................. 75

4-7 Probing experiments measuring the normalized joint entropy of elevation and luminance for foliage regions and non-foliage regions. ...... 77

4-8 Probing experiments measuring the normalized joint entropy of elevation and luminance for foliage regions and non-foliage regions. ...... 78

4-9 Probing experiments measuring the normalized joint entropy of elevation and luminance for foliage regions and non-foliage regions. ...... 79

4-10 Probing experiments measuring the normalized joint entropy of elevation and luminance for foliage regions and non-foliage regions. ...... 80

4-11 Bar chart for values given in Table 4.1. The error bars indicate the minimum and maximum values. ......................... 82

4-12 Bar chart for values given in Table 4.2. The error bars indicate standard deviations. ........................................... 83

4-13 Bar chart for values given in Table 4.4. The error bars indicate the minimum and maximum values. ......................... 85

4-14 Bar chart for values given in Table 4.5. The error bars indicate standard deviations. ........................................... 86

4-15 Example of registration using an initial perturbation from Table 3.2. Registration was performed by measuring the joint entropy between luminance and elevation for non-foliage regions. .......... 88

5-1 Depiction of Delaunay triangulation process for inferring a 3D surface from LIDAR point cloud. ............................... 91
5-2 Demonstration of texture mapping showing LIDAR point cloud (a), Delaunay mesh (b), and texture mapped mesh (c). The blue regions in (c) indicate areas where the mesh could not be textured due to occlusion. ................................................................. 93

5-3 Screenshot from multi registration algorithm. Blue pixels indicate occluded regions or background regions. These pixels are not used in the registration algorithm. ................................................................. 95

5-4 Probing experiments for Image 1 and Image 2 measuring the normalized joint entropy with the two attributions described in this chapter and the non-foliage method from the previous chapter. ............. 99

5-5 Probing experiments for Image 3 and Image 4 measuring the normalized joint entropy with the two attributions described in this chapter and the non-foliage method from the previous chapter. ............. 100

5-6 Probing experiments for Image 5 and Image 6 measuring the normalized joint entropy with the two attributions described in this chapter and the non-foliage method from the previous chapter. ............. 101

5-7 Probing experiments for Image 7 and Image 8 measuring the normalized joint entropy with the two attributions described in this chapter and the non-foliage method from the previous chapter. ............. 102

5-8 Bar chart for values given in Table 5.1. The error bars indicate the minimum and maximum values. ................................................................. 104

5-9 Bar chart for values given in Table 5.2. The error bars indicate standard deviations. ................................................................. 105
List of Tables

3.1 Range of camera parameter perturbations. These are referred to as the small range perturbations. ........................................ 52
3.2 Large range camera parameter perturbations ........................................ 55
3.3 Number of correctly registered images (out of 100) randomly sampled perturbations for small range parameters. .................. 59
3.4 Mean registration times in seconds (for correctly registered images). The times correspond to tests from the small range parameters. .... 60
3.5 Mean number of iterations (for correctly registered images). The number of iterations corresponds to tests from the small range parameters. 61
3.6 Number of correctly registered images (out of 100) randomly sampled perturbations for large range parameters. .................. 62
3.7 Mean registration times in seconds (for correctly registered images). The times correspond to tests for the large range parameters. ...... 63
3.8 Mean number of iterations (for correctly registered images). The number of iterations corresponds to tests for the large range parameters. 64
4.1 Number of correctly registered images (out of 100) for the small range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter. ............................. 82
4.2 Mean registration times in seconds (for correctly registered images). The times correspond to tests from the small range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter. ....................... 83
4.3 Mean number of iterations (for correctly registered images). The times correspond to tests from the small range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter. .............................. 84

4.4 Number of correctly registered images (out of 100) for the large range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter. .............................. 85

4.5 Mean registration times in seconds (for correctly registered images). The times correspond to tests from the large range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter. .............................. 86

4.6 Mean number of iterations (for correctly registered images). The times correspond to tests from the large range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter. .............................. 87

5.1 Number of correctly registered images out of 100 randomly sampled perturbations for multi and multi-non-folg attributions. Attributions from previous chapters are also shown. The perturbations were sampled from the large parameter range listed in Table 3.2 .............................. 104

5.2 Mean registration times in seconds for multi and multi-non-folg attributions. Attributions from previous chapters are also shown. Each image was tested with 100 randomly chosen perturbations. The perturbations were sampled from the large parameter range listed in Table 3.2 .............................. 105

5.3 Mean number of iterations for multi and multi-non-folg attributions. Attributions from previous chapters are also shown. Each image was tested with 100 randomly chosen perturbations. The perturbations were sampled from the large parameter range listed in Table 3.2 .............................. 106
Chapter 1

Introduction

Virtual reality 3D models are useful for understanding a scene of interest. 3D models are valuable for applications such as urban planning and simulation, interpretation of reconnaissance data, and real-time emergency response. Urban 3D modeling has also gained popularity in entertainment and commercial applications, and has been implemented with geographical image libraries such as Google Earth and Live Search Maps [12, 20]. Models are constructed by texture mapping aerial and ground images onto 3D geometry models of the scene. While geometry models have traditionally been constructed manually, recent advances in airborne laser radar (LIDAR) imaging technology have made the acquisition of high resolution digital elevation data more efficient and cost effective.

One challenge in creating such models is registering the 2D optical imagery with the 3D LIDAR imagery. This can be formulated as a camera pose estimation problem where the transformation between 3D LIDAR coordinates and 2D image coordinates is characterized by camera parameters such as position, orientation, and focal length. Consistent with previous literature, we assume that a coarse approximate of camera parameters is provided [31, 10, 5]. This is often the result of imperfect GPS/INS systems or a user selected orientation. Precise manual camera pose selection is difficult as it requires simultaneous refinement of numerous camera parameters. Registration can be achieved more efficiently by manually selecting pairs of correspondence points, but this task becomes laborious for situations where many images must be registered.
to create large 3D models. Some methods have been developed for performing automatic registration, but they suffer from being computationally expensive and/or demonstrating low accuracy rates [31, 10, 5]. In this thesis, we present a variety of novel methods for performing fast and accurate automatic registration.

Many aspects of this problem make it different from other registration problems. The most obvious is the nature of the 3D point cloud. Unlike traditional volumetric data, where intensity values are defined densely over 3D voxels, LIDAR data is given as a collection of ungridded 3D coordinates. Furthermore, the LIDAR point cloud does not provide intensity values, which are necessary for many conventional registration procedures. Last, the LIDAR data does not directly provide occlusion reasoning, meaning that correspondence between returns and physical surfaces is not explicitly given. This is shown in Figure 3-1 (b), where points from behind the building are visible as if the building wall were transparent. Thus, one must infer occlusions by making assumptions about the geometric properties of the data (we accomplish this with a Delaunay triangulation, described in Chapter 5).

In this thesis, we discuss a novel methodology for performing automatic camera pose estimation wherein we exploit the statistical dependency between LIDAR features and optical appearance in urban scenes. We start with basic registration methods that utilize LIDAR elevation, LIDAR probability of detection values, and optical luminance values. Next, we show how foliage detection can be used in conjunction with the basic registration methods to improve registration accuracy. Last, we consider registration scenarios where an overlapping optical image has already been registered with the LIDAR data, and the existing registered image is used to perform additional registrations. All of our algorithms are implemented with OpenGL, providing registration times on the order of seconds (in contrast to existing methods, which run on the order of minutes or longer [31, 10, 5]). We compare the performance of the registration algorithms and discuss the trade-offs associated with different types of images.

---

1Some datasets have probability of detection values, as we describe later
In summary, this thesis makes the following contributions:

- Empirical observation that LIDAR elevation and optical luminance are statistically dependent.
- Fast implementation of mutual information based registration for LIDAR and optical images.
- Use of foliage detection for improving registration reliability.
- Method for employing an existing registered optical image to improve registration reliability.
- Comparison of registration algorithm performance.

The organization of the thesis is as follows. Chapter 2 begins with a discussion of previous work. It describes existing methods for multi-view registration, single-view registration, and foliage detection. This is followed with an introduction to airborne LIDAR, a description of the camera model and parameters used in the registration algorithms, the method for obtaining ground truth registration, and an overview of mutual information based registration.

In Chapter 3, we present attribution methods for measuring mutual information between optical and LIDAR imagery. We analyze the objective landscape for the mutual information measures with a series of probing experiments, which is important for optimization routines. We then perform registration experiments with randomly chosen starting parameters and compare the performance of the different attribution methods.

Chapter 4 presents a method for using LIDAR foliage detection to improve registration. We begin with a description of the detection method and then analyze its performance on our data. Next, we analyze the joint elevation-luminance properties of foliage and non-foliage points, and show that by excluding foliage points, registration improves significantly. Since we focus on urban scenes, there is a significant amount of foliage regions, but they do not dominate the scenes. The improvement by
excluding foliage points is demonstrated with probing experiments and registration tests, which are compared to the methods described in Chapter 3.

Chapter 5 considers the scenario where an overlapping optical image is already registered with the LIDAR data, and presents a technique for draping the existing image onto the LIDAR data to perform registration. We show two methods for utilizing the draped image. Again, probing experiments and registration tests are presented to compare the method with the methods from other chapters. We ultimately find that use of an existing registered image gives the most reliable registration in comparison to the other methods presented.

A summary and conclusion are presented in Chapter 6 with a discussion of future work. The appendices provide proofs and algorithm pseudocode.
Chapter 2

Background

2.1 Previous Work

There has been a considerable amount of research in registering multi-view optical images with LIDAR imagery and other geometric models. Liu et al. [18] applied structure-from-motion to a collection of photographs to infer a sparse set of 3D points, and then performed 3D-3D registration. While 2D-3D registration is considered, their work emphasizes 3D-3D registration. Zhao et al. [34] used stereo vision techniques to infer 3D structure from video sequences, followed by 3D-3D registration with the iterative closest point (ICP) algorithm. Both of these methods demonstrate notable results with little or no prior camera information, such as global positioning system (GPS) data, but they require many overlapping images of the scene of interest.

In the area of single-view registration, Vasile et al. [31] used LIDAR data to derive a pseudo-intensity image with shadows for correlation with aerial imagery. They used a gradient based method for inferring walls in the LIDAR data. Their registration procedure starts with GPS and camera line of sight information and then uses an exhaustive search over translation, scale, and lens distortion. Frueh et al. [10] developed a similar system based on detection and alignment of line segments in the optical image and projections of line segments from the 3D image. Using a prior camera orientation with accuracy comparable to that of a GPS and inertial navigation system (INS), they used an exhaustive search over camera position, orientation, and
focal length. Their system requires approximately 20 hours of computing time on a standard computer. Both methods demonstrate accurate registration results, but are computationally expensive.

There are a variety of algorithms that utilize specific image features to perform registration. Troccoli and Allen [30] used matching of shadows to align images with a 3D model. This requires a strong presence of shadows as well as knowledge of the relative sun position when the photographs were taken. Kurazume et al. [16] used detection and matching of edges for registration. One drawback of this method is that it requires a relatively dense 3D point cloud to infer edges. Stamos and Allen [29] used matching of rectangles from building facades for alignment. Yang et al. [33] used feature matching to align ground images, but they worked with a very detailed 3D model. These methods are not robust for all types of urban imagery, and are not well suited for sparse point clouds.

Other approaches have employed vanishing points. Lee et al. [17] extracted lines from images and 3D models to find vanishing points. Their system cannot register all types of imagery, as it was designed for ground-based images with clearly visible facades. Ding et al. [5] used vanishing points with aerial imagery to detect corners in a similar manner, and used M-estimator sample consensus to identify corner matches. Starting with a GPS/INS prior, their algorithm runs in approximately 3 minutes, but only achieves a 61% accuracy rate for images of a downtown district, a college campus, and a residential region. Liu and Stamos [19] used vanishing points and matching of features to align ground images with 3D range models. All of these approaches are dependent on the strong presence of parallel lines to infer vanishing points, which limits their ability to handle different types of imagery.

To our knowledge, there are no methods that utilize foliage detection for registration, but there has been some research in foliage detection with LIDAR. Carlberg et al [2] classified urban landscapes using binary classifiers, 3D shape analysis, and region growing. The algorithm identifies water, ground, roof, and trees. The main drawback of their algorithm is that it requires color LIDAR data. Roggero [28] presents an algorithm that uses local plane fitting and variance propagation to classify LIDAR points.
as ground, vegetation, buildings, or outliers. In contrast to the method of Carlberg et al, it does not require color or intensity values for the LIDAR point cloud.

2.2 Airborne LIDAR

Airborne light detection and ranging (LIDAR), also known as light detection and ranging (LADAR), is a relatively new technology for collecting 3D aerial imagery. A detailed description of LIDAR systems is beyond the scope of this thesis, but can be found in the references [22, 6, 13]. Basic airborne LIDAR systems operate by scanning the ground plane with a laser and measuring the return delay of the laser pulses. This effectively measures the range of the ground locations scanned. The sensor has a highly precise global positioning system (GPS) and inertial navigation system (INS) that tracks the relative 3D positions of the returned pulses. The resulting dataset is a collection of 3D coordinates. The post-collection data often requires extensive processing to filter noise and achieve a meaningful 3D image. While the final 3D image may still have some noise, we assume that these errors are negligible in comparison to the uncertainty of the registration parameters. Some LIDAR sensors measure probability of detection values, or pdet values, which are derived from the number of photons received at each 3D coordinate. Large datasets are often collected by flying over collection regions in a raster pattern. The final post-processed datasets are usually ungridded and have noticeable overlapping swaths from this collection method.
Figure 2-1: Collection method for airborne LIDAR. The scene is scanned with a laser to measure range (shown on the left), producing a point cloud (shown on the right).
2.3 Camera Model

The camera model we use is the finite projective camera described in [14]. The transformation from 3D homogeneous coordinates to 2D homogeneous coordinates is given by the $3 \times 4$ matrix

$$T = KR[I \mid -C]$$

(2.1)

where $C = [C_x, C_y, C_z]^T$ is the camera center, $I$ is the identity matrix, and $R$ is the rotation matrix describing the orientation of the camera. $R$ is given by the product of rotation matrices

$$R = \begin{bmatrix}
c \gamma c \beta & c \alpha s \gamma + s \alpha c \gamma \beta & s \gamma \alpha - c \gamma c \alpha s \beta \\
-c \beta s \gamma & c \alpha c \gamma - s \alpha s \gamma s \beta & s \alpha c \gamma + c \alpha s \gamma s \beta \\
s \beta c & s \alpha c \beta & c c \beta 
\end{bmatrix}$$

(2.2)

where $\alpha$, $\beta$, and $\gamma$ are the Euler angles describing yaw, pitch, and roll. The notation $c$ indicates cosine while $s$ indicates sine (e.g. $c \alpha = \cos(\alpha)$). The matrix $K$ is the camera calibration matrix and has the form

$$K = \begin{bmatrix} f_x & t & x_0 \\
0 & f_y & y_0 \\
0 & 0 & 1 \end{bmatrix}$$

(2.3)

where $f_x$ and $f_y$ are the focal lengths in the $x$ and $y$ directions, $(x_0, y_0)$ are the coordinates of the principal point, and $t$ is the skew. The principal point indicates the location of the center of the image on the image plane and the skew determines the angle between the image plane $x$- and $y$- axis. For our images, the skew and principal point parameters are unnecessary, yielding $t, x_0, y_0 = 0$. Additionally, $f_x = f_y$ under the assumption of square pixels. This leaves a constrained finite projective camera parameterized by seven variables: $C_x, C_y, C_\alpha, \beta, \gamma$, and the field-of-view (fov). The fov can be computed from the focal length. We sometimes refer to $C_x, C_y,$
and $C_z$ as $x$, $y$, and $z$, respectively; the definitions should be clear based on context. A diagram of the camera model is shown in Figure 2-2.

Figure 2-2: Geometric diagram of camera model
2.4 Ground Truth Registration

Expert chosen correspondence points are utilized to determine ground truth. Correspondence points are chosen by identifying salient geometric features visible in both images. We use the algorithm given by [14, p. 184] to determine the best camera matrix. The algorithm minimizes the sum of squared algebraic errors between correspondence points in the image plane,

\[
T_G = \arg \min_T \sum_i d_{\text{alg}}(X'_i, TX_i)
\]  

(2.4)

where \(X'\) represents a 2D homogeneous point, \(X\) represents a 3D homogeneous point, and \(i\) is the index for each point. To define the algebraic distance, we introduce some new notation. Let \(X' = (x', y', w')^T\) and \(\hat{X}' = TX = (\hat{x}', \hat{y}', \hat{w}')^T\) under the conventional definition of 2D homogeneous coordinates. The algebraic distance is then defined as

\[
d_{\text{alg}}(X', \hat{X}')^2 = (y'\hat{w}' - w'\hat{y}')^2 + (w'\hat{x}' - x'\hat{w}')^2.
\]  

(2.5)

For comparison, we define the conventional geometric distance:

\[
d_{\text{geo}}(X', \hat{X}') = \left( \frac{(x'w' - \hat{x}'\hat{w}')^2 + (y'w' - \hat{y}'\hat{w}')^2}{w'w'} \right)^{1/2} \]

(2.6)

This shows the algebraic distance is somewhat related to geometric distance [14]. Unfortunately, there is no graphical description of algebraic distance. While the algorithm can be used with either algebraic or geometric distance, we found that results with the algebraic distance were slightly more accurate. We also determined that 30 correspondence points were sufficient for a precise registration. All ground truth registrations described in this thesis were registered with this method and 30 points. Figure 2-3 shows an example of an optical image and a LIDAR image marked with 30 correspondence points.
Figure 2-3: Optical image (a) and LIDAR image (b) with 30 manually selected correspondence points
2.5 Mutual Information Registration

Statistical and information-theoretic methods have been used extensively for multi-modal registration of medical imagery. Methods based on these principals have demonstrated excellent performance for a wide variety of 2D-2D and 2D-3D registration applications, e.g. [35, 36]. The methods were originally proposed contemporaneously by Viola and Wells [32] and Maes et al [21]. Since their original proposal, these methods (and variations thereof) have become the standard method for automatic registration of dense volumetric medical imagery (e.g. CT and MRI). As these algorithms use grayscale intensity values to evaluate statistics, there is not a direct way to apply them to raw LIDAR and luminance values. Consequently, we attribute features to both types of imagery and evaluate registration statistics over the features. This is discussed in detail in the following chapters.

Since our problem involves registration of imagery in two dimensions with a point cloud in three dimensions, we evaluate registration statistics in the 2D image plane via projection of the LIDAR features within the constraints of a camera model for comparison with the image features. We define $u(x, y)$ and $v(x, y)$ as the the image features and projected LIDAR features, respectively, on the x-y image plane such that the images are correctly registered. We denote $v_o$ as the initial unregistered projection of LIDAR features obtained from a user selected pose approximation or GPS/INS data. For a specific camera matrix $T$ (defined in Section 2.3), the projected LIDAR features are given by $v_T$.

Mutual information (MI) based registration methods seek the camera matrix that maximizes the MI between the distribution of photograph features and projected LIDAR features:

$$T_{MI} = \arg\max_T I(u; v_T).$$  \hspace{1cm} (2.7)

One definition of mutual information is the Kullback-Leibler (KL) divergence of the joint distribution and the product of marginals

$$I(u; v_T) = D(p(u, v; T)||p(u)p(v; T)),$$  \hspace{1cm} (2.8)
where \( p(u, v; T) \) and \( p(v; T) \) are probability density functions of features for a camera matrix \( T \). Consequently, maximizing MI is equivalent to maximizing the KL divergence between the joint distribution under evaluation and the case where the images are independent. This inherently assumes that the images are best aligned when their statistical dependence is high. KL divergence, defined as an expectation, can be approximated from a sample histogram as

\[
I(u; T) = \int \int p(u, v; T) \log \left( \frac{p(u, v; T)}{p(u)p(v; T)} \right) dudv \tag{2.9}
\]

\[
\approx \sum_{i=1}^{N_u} \sum_{j=1}^{N_v} \hat{p}(u_i, v_j; T) \log \left( \frac{\hat{p}(u_i, v_j; T)}{\hat{p}(u_i)\hat{p}(v_j; T)} \right) \tag{2.10}
\]

where \( \hat{p}(\cdot) \) denotes a marginal or joint histogram estimate of a density and \( N_u \) and \( N_v \) denote the number of distinct bins for each modality. See [36] for a detailed discussion of these approximations.

Mutual information can also be expressed in terms of entropies of the LIDAR features, optical features, and their joint entropy:

\[
I(u; v_T) = H(u) + H(v_T) - H(u, v_T). \tag{2.11}
\]

In our case, the entropy of the image features remains constant, and the entropy of the LIDAR features remains approximately constant for small perturbations. Accordingly, the calibration matrix that minimizes the joint entropy is a sufficient approximation for the calibration matrix that maximizes the mutual information. That is, minimizing

\[
H(u; v_T) \approx \sum_{i=1}^{N_u} \sum_{j=1}^{N_v} \hat{p}(u_i, v_j; T) \log \left( \hat{p}(u_i, v_j; T) \right) \tag{2.12}
\]

is equivalent to maximizing MI over \( T \). A large portion of our analysis in the following chapters will strictly use joint entropy as a similarity measure rather than the full MI evaluation.
2.6 Quasiconvexity

We use downhill simplex optimization in conjunction with MI measures throughout the thesis. For downhill simplex optimization, quasiconvexity over parameter ranges is important for finding the true optima and is used to define the basin of attraction.

A function $f : \mathbb{R}^n \to \mathbb{R}$ is quasiconvex if its domain and all sublevel sets

$$L_k = \{x \in \text{dom} f | f(x) \leq k\}, \quad (2.13)$$

for $k \in \mathbb{R}$, are convex [1]. All convex functions are quasiconvex, though the converse is not true.
Chapter 3

Basic Image Registration

In this chapter, we present a basic procedure for optical-LIDAR image registration. We begin by deriving three methods for attributing the datasets based on the elevation values and probability of detection values in the LIDAR data, and luminance in the optical image. We describe the intuition behind these attributions and demonstrate empirically how they measure dependence between image modalities.

Next, we perform a series of probing experiments to explore the mutual information landscape based on the three attribution methods. Since our intent is to use the MI measures in conjunction with a search-based optimization routine, it is important to verify that the objective function is smooth and quasiconvex. We perform probing experiments using eight images and find that these properties hold over a sufficient range of camera parameters.

We then implement a registration algorithm based on the three attributions for measuring MI. The algorithm is implemented in OpenGL, which allows for fast geometric manipulation of the LIDAR point cloud and thus fast registration times. Downhill simplex optimization is employed, so an analytical expression of the derivative is not necessary.

The chapter is completed with a series of registration tests. For each of eight test images, we generate 100 random camera perturbations to evaluate the registration algorithm. We analyze the post-algorithm registration results and find accuracy rates for all three attribution methods to be above 90% for a certain range of camera
parameter perturbations. The registration run times are also measured in these tests. On average, images are registered in less than 30 seconds for all three attribution methods.

### 3.1 Attribution Methods

An important issue in using information-theoretic methods for our problem is attributing each data set with features/labels that are conducive to registration. Examples of features might be the local variance for each point in the LIDAR or the gradient of each pixel in the image. In order to be useful for registration, the features must provide an objective function with a minimum at the correct registration value (as a function of camera parameters) as well as a smooth quasiconvex shape. This shape is necessary so that an optimization algorithm can search for the correct registration without converging to a local minimum. We present three different attribution methods in this section: one based on LIDAR elevation and optical luminance, one based on LIDAR probability of detection and optical luminance, and one based on LIDAR elevation, probability of detection, and optical luminance.

#### 3.1.1 Elevation

The first data attribution is based on the observation that the visual appearance of urban scenes tend to vary in a structured way by elevation for architectural reasons. This suggests that there is a measurable dependence between the optical appearance and the LIDAR elevation values. This can be deduced intuitively in viewing Figure 3-1, which shows a scene depicted by both modalities. It is clear that similar structure exists across both images. In particular, the street and grass are at a low elevation shown in purple, while the brown building roofs shown in red, yellow, and white are at a higher elevation. Using only a region of the pair of images, one could make an informed prediction about the appearance of another region in the optical image based on elevation information.

While this dependence makes sense intuitively, it is important to analyze the
dependence empirically. For a preliminary analysis, we look at the joint probability mass function (PMF) of the LIDAR elevation values and the image luminance values for a registered image. We use image luminance (instead of three-channel color) here and throughout the thesis to reduce the necessary number of labels from $2^{24}$ to $2^8$. Image luminance is given by a weighted sum of red, green, and blue pixel values:

$$L = 0.2989R + 0.5870G + 0.1140B.$$  \hspace{1cm} (3.1)

Figure 3-2 (a) shows an elevation-encoded projection of the LIDAR point cloud that was registered with the optical image using the method described in 2.4. The point cloud was rendered in OpenGL with grayscale intensities indicating different elevations, where brighter is higher. These intensities were discretized to 256 levels with a simple linear mapping based on the minimum and maximum LIDAR elevation values. The joint PMF for the LIDAR image and corresponding optical luminance image is shown in Figure 3-3, along with the product of marginals PMF. The images used to generate the PMFs are shown in Figure 3-2. The blue regions in Figure 3-2 (a) indicate areas where no LIDAR points were projected into the 2D image plane, and are considered background regions. These pixels and the corresponding luminance values in the optical image were not used in generating the PMF figure.

The joint PMF shows a complex, yet structured relationship between luminance and elevation. The two values are clearly not independent because the PMF differs from the product of marginals shown in Figure 3-3 (b). The PMF is difficult to characterize as it is multi-modal and does not resemble any conventional PMFs (e.g. a Gaussian distribution). This motivates the use of mutual information to describe the dependence of the two variables. It is important to note that the PMF is unique to the two selected images and does not necessarily characterize the joint luminance-elevation distribution of the entire scene or other scenes. It does, however, provide a discretized estimate of the joint distribution. A direct calculation of the mutual information (MI) from the PMF yields 0.3795 bits (with a maximum possible MI of 5.2064 bits), verifying that there is measurable dependence.
Figure 3.1: Optical image (a) and LIDAR image (b) of an urban scene. The different colors in the LIDAR image indicate different elevations.

Although there is nonzero mutual information between the elevation and luminance values, this does not mean it is sufficient for a registration cost function. The mutual information must smoothly decrease as the camera parameters are perturbed from their correct values. This is necessary for using search-based optimization techniques, though not important if an exhaustive search is used (we only consider search-based optimization since it is significantly faster). To provide an exaggerated test of this phenomena, we look at the joint PMF and MI of an image that is not correctly registered. This is shown in Figure 3.4. The two images are of the same scene, but the LIDAR image is drastically perturbed from the correct registration. The joint PMF still exhibits some structure, but has less structure than the one for the registered image. The measure of mutual information supports this observation, as it is only 0.1431 bits, compared to 0.3795 bits for the registered image. For both PMFs the maximum MI is 5.2064 bits. This suggests that the MI measure may be suitable for registration. However, this does not verify that the MI varies smoothly as camera parameters are perturbed; we analyze this issue later in the chapter.
Figure 3-2: LIDAR elevation image (a) and optical luminance image (b) used to generate the PMFs in Figure 3-3. The blue regions in (a) indicate areas where no LIDAR points were projected into the 2D image plane.
Figure 3-3: Log probability mass function (PMF) for the joint LIDAR elevation values and optical luminance values (a) and product of marginals (b). The mutual information calculated from the PMF is 0.3795 bits with a maximum MI of 5.2064 bits.
Figure 3-4: Log probability mass function (PMF) of the LIDAR elevation values and optical luminance values (c) for the images shown in (a) and (b). The images are of the same scene, but have not been registered to illustrate the lower mutual information. The blue regions in (a) indicate areas where no LIDAR points were projected into the 2D image plane. The mutual information calculated from the PMF is 0.1431 bits with a maximum MI of 5.2064 bits.
3.1.2 Pdet

The second attribution method that we consider uses probability of detection (pdet) values from the LIDAR sensor. Many LIDAR sensors measure the number of photons returned for each point in addition to the 3D coordinate. Viewing the LIDAR point cloud with these values often makes it much easier to understand the scene since the values look similar to grayscale intensity values. Since these values are measured by the sensor, they are already registered with the elevation values. Our particular data set has 8 bit precision for the pdet values, yielding 256 different intensity levels. A pdet encoded rendering of the point cloud is shown in Figure 3-5 next to the corresponding optical image luminance; the two images have been registered.

It is immediately apparent that certain features are visible in the pdet-encoded LIDAR that are not in the elevation-encoded LIDAR. For example, Figure 3-5 (b) shows walkways between the buildings that are not visible in Figure 3-1 (b). There is a more recognizable correspondence between the optical image luminance and pdet LIDAR than the luminance and elevation LIDAR. This is expected since the former are both derived in some sense from scene reflectivity. As with the previous attribution method, it is useful to view the joint PMF of the pdet values and the optical image luminance. This is presented in Figure 3-7, again with the product of marginals PMF. The images used to generate the PMFs are shown in Figure 3-6.

The joint PMF again shows a structured multi-modal relationship. It is interesting
to note that there is not a linear correlation between the luminance values and the pdet values, as one might expect since they are both based on scene reflectivity. There is some evidence of this in Figure 3-5. In particular, the walkways between the buildings are lighter than the grass in the optical image, yet darker than the grass in the pdet image. The mutual information measured from the PMF is 0.2539 bits with a maximum MI of 7.473 bits. Interestingly, this is smaller than the MI measured with the elevation measure (both are based on the same scene). One would expect the pdet-luminance MI to be higher since pdet and luminance are in some sense derived from scene reflectivity. However, it seems that the pdet values vary significantly based on material properties. This is evident by the fact that there are many more bright pdet values than luminance values (see Figure 3-6). Additionally, in other scenes where water is present, the LIDAR returns are nonexistent. We calculated the mutual information with a non-registered image. We do not show a figure, but the non-registered mutual information is 0.0942 bits with a maximum MI of 7.4163 bits. These properties indicate again that mutual information may be a good cost function for a registration algorithm.
Figure 3-6: LIDAR pdet image (a) and optical luminance image (b) used to generate the PMFs in Figure 3-7. The blue regions in (a) indicate areas where no LIDAR points were projected into the 2D image plane.
Figure 3-7: Log probability mass function (PMF) for the joint LIDAR pdet values and optical luminance values (a) and product of marginals (b). The mutual information calculated from the joint PMF is 0.2539 bits with a maximum of 7.473 bits.
3.1.3 Dual

The last attribution method that we consider combines both LIDAR elevation and pdet with image luminance. We note that the elevation values and pdet values are already registered. The most straightforward way to use all three features is by measuring the mutual information between the image luminance values, and the LIDAR elevation and pdet values

\[ I(v_e, v_p; u), \]

where \( u \) is the image luminance, \( v_e \) is the LIDAR elevation, and \( v_p \) is the LIDAR pdet values. This can be expressed in terms of joint and marginal entropies:

\[ I(v_e, v_p; u) = H(v_e, v_p) + H(u) - H(u, v_e, v_p). \]  (3.2)

Estimating \( H(u, v_e, v_p) \) becomes problematic due to an insufficient number of samples to fill a histogram, assuming that we do not want to discretize the variables to fewer than 256 levels. The reasoning is as follows: all three features have 256 levels, meaning the histogram would have \( 256^3 \) or approximately 16 million distinct bins. The images that we use have fewer than 700,000 pixels, so many histogram bins would be left empty while others would only have one or two counts. We circumvent this problem by approximating the LIDAR pdet values as statistically independent of the elevation values conditioned on the optical image luminance values. The graphical model representing this approximation is shown in 3-8. One interpretation of this assumption is that once the optical image luminance is known for a particular pixel, knowing the LIDAR pdet does not provide any additional information for estimating the LIDAR elevation of the pixel.

Figure 3-8: Graphical model describing conditional independence relationship of LIDAR elevation, LIDAR pdet, and image luminance.
The approximation leads to the following approximation of the joint entropy term:

\[ H(u, v_e, v_p) = H(u, v_e) + H(u, v_p) - H(u). \]  \hspace{1cm} (3.3)

A proof of this statement is given in Appendix A.1.

3.2 Probing Experiments

As mentioned in the previous section, one crucial requirement for a registration cost function is that it is relatively smooth and quasiconvex near the correct registration value. This is necessary so that search-based optimization algorithms can find the correct minimum value without getting trapped in local minima. In this section, we analyze the shape of the objective function with a series of probing experiments for the three attribution methods.

Since we are interested in optimizing over multiple camera parameters, it is desirable to fully view the multivariate objective function near the registered value. Unfortunately this is infeasible as it would require viewing a 7-dimensional diagram. We instead look at each parameter individually. We set all parameters at their correct values (determined by the ground truth registration) and vary each parameter to view the shape of the objective function. While this does not allow us to fully characterize the cost function, it gives a general indication of its value for registration.

We perform probing experiments with eight images that are shown in Figure 3-9. The images were provided by Pictometry International [25]. We chose these images with the intent of testing a diversity of scenes available in our data set. The first three images are different perspectives of a college campus with buildings and some foliage. Image four was selected because it has few rectangular structures and it is relatively flat. Image seven differs from the other images in that the buildings are much smaller overall.

The probing experiments are shown in Figures 3-10, 3-11, 3-12, and 3-13. In all probing experiments, some of the angle parameters \((\alpha, \beta, \text{ and the field of view})\)
were varied over only two degrees. These angle perturbations seem small, but it is important to note that the long standoff distance magnifies the effect of the Euler angles (except for $\gamma$) and the fov angle. This is demonstrated in Figure 3-14, which shows a blend of the LIDAR data and the corresponding image with an $\alpha$ perturbation of one degree.

The probing experiments demonstrate generally reliable smoothness and convexity over a subset of the displacement parameters shown. The pdet cost functions are somewhat more jagged than the other two. This is visible with almost all of the images, thought it is particularly noticeable with Image 8. We hypothesize that this is due to the fact that there is more high frequency spatial variation in the pdet values than the elevation values.

While the elevation, pdet, and dual attributions show fairly similar objective function shapes, they are notably different for Image 4. With this image, the elevation attribution is quasiconvex over a smaller range of parameters in comparison to the pdet attribution. This is to be expected since Image 4 is the test image with the least amount of elevation variation. Since there is more variation in the reflectivity of the image (e.g. the baseball field), the pdet attribution performs better.

Analyzing these diagrams while keeping in mind that they do not fully characterize the 7-dimensional objective function, we decided on a range of parameter parameter displacements near the perceived basin of attraction. These displacement values are shown in Table 3.1. We use these displacement values to run registration experiments in the next section.
Figure 3-9: Images used for probing experiments and registration experiments
Figure 3-10: Plots of normalized joint entropy for Image 1 and Image 2. Each camera parameter is smoothly varied while the other parameters are held constant at their ground truth values.
Figure 3-11: Plots of normalized joint entropy for Image 3 and Image 4. Each camera parameter is smoothly varied while the other parameters are held constant at their ground truth values.
Figure 3-12: Plots of normalized joint entropy for Image 5 and Image 6. Each camera parameter is smoothly varied while the other parameters are held constant at their ground truth values.
Figure 3-13: Plots of normalized joint entropy for Image 7 and Image 8. Each camera parameter is smoothly varied while the other parameters are held constant at their ground truth values.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_x$</td>
<td>20</td>
<td>meters</td>
</tr>
<tr>
<td>$C_y$</td>
<td>20</td>
<td>meters</td>
</tr>
<tr>
<td>$C_z$</td>
<td>20</td>
<td>meters</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.5</td>
<td>degrees</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.5</td>
<td>degrees</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>5</td>
<td>degrees</td>
</tr>
<tr>
<td>fov</td>
<td>0.5</td>
<td>degrees</td>
</tr>
</tbody>
</table>

Table 3.1: Range of camera parameter perturbations. These are referred to as the small range perturbations.

Figure 3-14: Fade between LIDAR image and optical image with an $\alpha$ perturbation of one degree.
3.3 Registration Algorithm

The automatic registration algorithm that we present is based on a nonlinear optimization over camera parameters, and utilizes the cost functions described earlier. The algorithm renders 3D points that are projected into the image plane for evaluating statistics (in a similar fashion to the method used to make Figure 3-3). The point cloud is rendered in OpenGL with a point size of 1.0, meaning that each projected point appears as one pixel in the image plane. Only image pixels with a corresponding projected LIDAR point are used for MI calculation; the others are considered background points and discarded. The point cloud is rendered numerous times for each iteration. Downhill simplex optimization is used, which does not require an analytical derivative expression [24].

There are several benefits to using OpenGL in the registration process. First, OpenGL automatically performs occlusion reasoning on a point by point basis. This is beneficial, but it is not a perfect process since surface regions where there are no point returns appear transparent. Next, using OpenGL in conjunction with advanced graphics hardware allows us to place the entire LIDAR point cloud on the graphics card memory, making 3D rendering extremely efficient. Specifically, the OpenGL implementation uses vertex buffer objects, which allows for the 3D LIDAR coordinates and intensity values (representing elevation or pdet) to be uploaded onto the graphics card. The 3D coordinates do not change for each rendering iteration. Rather, the OpenGL camera projection matrix changes during each iteration to calculate the 3D to 2D projection of points.

The efficient use of graphics memory is important for large LIDAR point clouds. We have one LIDAR tile for each optical image in our dataset. Each tile is 1000 x 1000 meters and has approximately 5 million points. This yields a planar density of about 5 points per square meter. All of the optical images are 1002 x 668 pixels.
3.4 Registration Tests

We test our algorithm with the collection of eight urban images of Lubbock, Texas shown in Figure 3-9. As with previous work, it is conventional to start with an approximate initial registration that is available from a GPS/INS system or a user selected approximation. We simulate a variety of initial approximate registrations and characterize the accuracy of the final registration using the three different MI attributions. Overall, the results demonstrate that the MI measures are reliable for registration and that the algorithm is faster than previous algorithms by an order of magnitude.

For each of the eight images, we simulated 100 initial coarse registrations. The algorithm was tested over two ranges of parameter perturbations. Table 3.1 shows the range of camera parameter perturbations chosen to approximate the basin of attraction, which we refer to as the small range parameters. Table 3.2 shows a larger range of parameter perturbations, which we refer to as the large range parameters. The large range parameters are each twice the size of the small range parameters. Our images did not contain focal length information, so we included the field-of-view parameter in our parameter search. When the focal length is known, the complexity of the search is reduced to six parameters. For each range of camera parameters, we randomly sampled parameters uniformly from the range to obtain initial parameters for registration.

All registration tests were performed on a desktop PC with a dual core 2.13 GHz processor and 4 GB of RAM. The NVIDIA GeForce 9800 GX2 video card was used, which has 1GB of on-board dedicated memory. The registration algorithms were implemented in C++, utilizing both QT and QGLViewer [27, 26].

It was apparent by simple inspection whether or not the images were correctly registered. In all cases, the image was either registered close enough for projective texture mapping or clearly unaligned. To quantitatively characterize this phenomena, we employed the correspondence points that were used for ground truth registration (as described in 2.4). We measured the mean squared error (MSE) of the correspon-
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_x$</td>
<td>40</td>
<td>meters</td>
</tr>
<tr>
<td>$C_y$</td>
<td>40</td>
<td>meters</td>
</tr>
<tr>
<td>$C_z$</td>
<td>40</td>
<td>meters</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>1.0</td>
<td>degrees</td>
</tr>
<tr>
<td>$\beta$</td>
<td>1.0</td>
<td>degrees</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>10</td>
<td>degrees</td>
</tr>
<tr>
<td>fov</td>
<td>1.0</td>
<td>degrees</td>
</tr>
</tbody>
</table>

Table 3.2: Large range camera parameter perturbations

dence points for the initial perturbations as well as the post-algorithm registration. Although algebraic error was used in the ground truth registration algorithm, we chose to use geometric error here since it is more intuitive. Figure 3-15 shows MSE histograms for the initial perturbations and post-algorithm registrations corresponding to the large parameter range. These figures were generated from tests for all eight images using the elevation attribution. Distributions of MSE for the other attribution methods looked similar. We chose a MSE of 50 pixels$^2$ for the threshold of a correct registration. This was chosen in conjunction with the large cluster shown in Figure 3-15 (b). We also verified that this threshold was low enough to produce texture-mapped renderings with negligible errors.

Results for the small range parameters (shown in Table 3.1) are shown in Tables 3.3, 3.4, and 3.5, describing the number of successes, duration of the optimization, and the number of iterations, respectively. As expected, the dual measure of MI (which uses LIDAR elevation and pdet) demonstrates the best results, with an overall accuracy of 98.5%. However, it is interesting to note that marginal benefit of using the pdet MI measure (95.8%) and the dual MI measure (98.5%) over the elevation MI measure (93.5%) is relatively small. Table 3.4 shows fast registration times, which all averaged to be less than 20 seconds. The dual registration times are approximately the sum of the elevation and pdet registration times, which is expected since two images must be rendered for each iteration with our implementation. An example of an initial and final alignment are shown in Figure 3-21.

The results for the large range parameters are shown in Tables 3.6, 3.7, and 3.8.
As expected, the registration accuracies are lower overall. The elevation and pdet methods perform relatively similar, with accuracies of 66.5% and 65.8%, respectively. The marginal utility provided by the dual method is much more pronounced here, with an accuracy of 75.6%. However, the average runtime is significantly larger: 27.44 seconds.

Our optical images were acquired near the same time as the LIDAR data set, but there are a few differences in structure between the two datasets in some images. The most obvious example of this is in Image 8, where a taller building and elevated walkway were built after the image was taken, but before the LIDAR data was acquired. This is shown in Figure 3-16. With all three of the attribution methods described, the difference in structure did not cause errors in the registration (for registrations considered correct by the MSE threshold).

Examples of post-registration models are shown in 3-22. These models were created by inferring a 3D mesh from the point cloud using Delaunay triangulation and texture mapping the registered image onto the mesh [3]. A detailed description of this process is provided in section 5.1.
Figure 3-15: Histogram of mean squared error for initial perturbations (a) and post-algorithm registrations (b). The initial perturbations were drawn from the large parameter range. The histograms were generated with samples for all eight test images.
Figure 3-16: Example of discrepancies in image modes due to different collection times. The registration algorithm achieves the correct registration (c) despite the difference.
Table 3.3: Number of correctly registered images (out of 100) randomly sampled perturbations for small range parameters.

<table>
<thead>
<tr>
<th>JE Measure:</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>100</td>
<td>93</td>
<td>100</td>
</tr>
<tr>
<td>Image 2</td>
<td>97</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>Image 3</td>
<td>83</td>
<td>100</td>
<td>96</td>
</tr>
<tr>
<td>Image 4</td>
<td>76</td>
<td>83</td>
<td>97</td>
</tr>
<tr>
<td>Image 5</td>
<td>97</td>
<td>97</td>
<td>99</td>
</tr>
<tr>
<td>Image 6</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Image 7</td>
<td>96</td>
<td>95</td>
<td>97</td>
</tr>
<tr>
<td>Image 8</td>
<td>99</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>93.5%</td>
<td>95.8%</td>
<td>98.5%</td>
</tr>
</tbody>
</table>

Figure 3-17: Bar chart for values given in Table 3.3. The error bars indicate the minimum and maximum values.
<table>
<thead>
<tr>
<th>JE Measure:</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>6.30</td>
<td>8.12</td>
<td>14.18</td>
</tr>
<tr>
<td>Image 2</td>
<td>6.80</td>
<td>7.14</td>
<td>13.39</td>
</tr>
<tr>
<td>Image 3</td>
<td>6.44</td>
<td>8.59</td>
<td>14.67</td>
</tr>
<tr>
<td>Image 4</td>
<td>5.76</td>
<td>5.79</td>
<td>10.88</td>
</tr>
<tr>
<td>Image 5</td>
<td>6.55</td>
<td>6.58</td>
<td>12.39</td>
</tr>
<tr>
<td>Image 6</td>
<td>6.19</td>
<td>7.13</td>
<td>11.89</td>
</tr>
<tr>
<td>Image 7</td>
<td>6.77</td>
<td>8.27</td>
<td>14.44</td>
</tr>
<tr>
<td>Image 8</td>
<td>5.99</td>
<td>6.53</td>
<td>12.40</td>
</tr>
<tr>
<td>Total</td>
<td>6.35</td>
<td>7.27</td>
<td>13.03</td>
</tr>
</tbody>
</table>

Table 3.4: Mean registration times in seconds (for correctly registered images). The times correspond to tests from the small range parameters.

Figure 3-18: Bar chart for values given in Table 3.4. The error bars indicate standard deviations.
<table>
<thead>
<tr>
<th>JE Measure:</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>114.8</td>
<td>115.9</td>
<td>112.1</td>
</tr>
<tr>
<td>Image 2</td>
<td>127.4</td>
<td>123.1</td>
<td>118.5</td>
</tr>
<tr>
<td>Image 3</td>
<td>116.7</td>
<td>119.9</td>
<td>115.0</td>
</tr>
<tr>
<td>Image 4</td>
<td>98.5</td>
<td>104.0</td>
<td>99.6</td>
</tr>
<tr>
<td>Image 5</td>
<td>112.4</td>
<td>116.6</td>
<td>112.9</td>
</tr>
<tr>
<td>Image 6</td>
<td>114.9</td>
<td>116.9</td>
<td>110.0</td>
</tr>
<tr>
<td>Image 7</td>
<td>118.4</td>
<td>115.2</td>
<td>110.5</td>
</tr>
<tr>
<td>Image 8</td>
<td>111.2</td>
<td>118.4</td>
<td>108.8</td>
</tr>
<tr>
<td>Total:</td>
<td>115.6</td>
<td>116.2</td>
<td>110.9</td>
</tr>
</tbody>
</table>

Table 3.5: Mean number of iterations (for correctly registered images). The number of iterations corresponds to tests from the small range parameters.
<table>
<thead>
<tr>
<th>JE Measure:</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>91</td>
<td>61</td>
<td>84</td>
</tr>
<tr>
<td>Image 2</td>
<td>49</td>
<td>78</td>
<td>74</td>
</tr>
<tr>
<td>Image 3</td>
<td>52</td>
<td>90</td>
<td>67</td>
</tr>
<tr>
<td>Image 4</td>
<td>31</td>
<td>29</td>
<td>52</td>
</tr>
<tr>
<td>Image 5</td>
<td>84</td>
<td>69</td>
<td>81</td>
</tr>
<tr>
<td>Image 6</td>
<td>98</td>
<td>87</td>
<td>99</td>
</tr>
<tr>
<td>Image 7</td>
<td>43</td>
<td>52</td>
<td>63</td>
</tr>
<tr>
<td>Image 8</td>
<td>84</td>
<td>60</td>
<td>85</td>
</tr>
<tr>
<td>Total</td>
<td>66.5%</td>
<td>65.8%</td>
<td>75.6%</td>
</tr>
</tbody>
</table>

Table 3.6: Number of correctly registered images (out of 100) randomly sampled perturbations for large range parameters.

Figure 3-19: Bar chart for values given in Table 3.6. The error bars indicate the minimum and maximum values.
Table 3.7: Mean registration times in seconds (for correctly registered images). The times correspond to tests for the large range parameters.

<table>
<thead>
<tr>
<th>JE Measure:</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>8.72</td>
<td>10.65</td>
<td>29.95</td>
</tr>
<tr>
<td>Image 2</td>
<td>8.21</td>
<td>9.76</td>
<td>29.46</td>
</tr>
<tr>
<td>Image 3</td>
<td>9.92</td>
<td>10.29</td>
<td>28.65</td>
</tr>
<tr>
<td>Image 4</td>
<td>7.23</td>
<td>8.47</td>
<td>24.56</td>
</tr>
<tr>
<td>Image 5</td>
<td>8.86</td>
<td>8.60</td>
<td>25.61</td>
</tr>
<tr>
<td>Image 6</td>
<td>8.07</td>
<td>11.42</td>
<td>26.42</td>
</tr>
<tr>
<td>Image 7</td>
<td>9.93</td>
<td>9.26</td>
<td>29.19</td>
</tr>
<tr>
<td>Image 8</td>
<td>7.91</td>
<td>7.66</td>
<td>25.65</td>
</tr>
<tr>
<td>Total</td>
<td>8.61</td>
<td>9.51</td>
<td>27.44</td>
</tr>
</tbody>
</table>

Figure 3-20: Bar chart for values given in Table 3.7. The error bars indicate standard deviations.
Table 3.8: Mean number of iterations (for correctly registered images). The number of iterations corresponds to tests for the large range parameters.
Figure 3-21: Example of registration results with image superimposed on projected LIDAR image. The perturbation was sampled from the small range parameters.
Figure 3-22: Texture mapped scenes made from post-algorithm registrations
Chapter 4

Foliage Detection

In the last chapter, we found that there is a benefit in using probability of detection (pdet) values in conjunction with elevation values. This was demonstrated with the dual attribution method, which outperformed the elevation and pdet methods. This is expected since the pdet values show additional scene detail that is not visible in elevation only renderings. It is additionally beneficial because it is derived from scene reflectivity, just as image luminance is. However, pdet values are not always provided with LIDAR data sets. This motivates the development of more robust registration techniques using only the LIDAR coordinates.

We present a novel attribution method that utilizes foliage detection in this chapter. The attribution methods in the previous chapter make the assumption that all samples are equally useful in performing registration, which is not necessarily true. The analysis in this chapter explores one method for determining which samples are more useful than others based on foliage detection. Foliage is detected in the 3D point cloud with a plane fitting technique, and these points are subsequently ignored in using the elevation-based registration method. We discuss the reasoning for this technique both intuitively and empirically. This method ultimately allows us to achieve registration accuracies that are higher than those using pdet and elevation as discussed in the previous chapter.
4.1 Detection Method

The method that we use for detecting foliage regions uses a local plane fitting technique followed by a chi-square goodness of fit test. The goodness of fit test was originally proposed by Roggero [28] for finding foliage in 3D point clouds. The goodness of fit test effectively tries to determine whether or not a cluster of points constitute a planar surface. Points that do not fit the plane criteria are considered foliage.

To estimate the local plane for a neighborhood of points, we use the method based on least squares and principal component analysis described in [15]. The local plane for point \( i \) is defined by the center \( o_i \) and the normal vector \( \hat{n}_i \). For each 3D point \( x_i \), we consider all the neighboring points that are within a distance \( r \) from the point, which are denoted as \( N(x_i) \). The center \( o_i \) is found by determining the centroid of \( N(x_i) \). The normal \( \hat{n}_i \) is calculated by forming the covariance matrix for \( N(x_i) \)

\[
M = \sum_{y \in N(x_i)} (y - o_i)(y - o_i)^T.
\]  
(4.1)

The normal \( \hat{n}_i \) is then given by the eigenvector of \( M \) that corresponds to the smallest eigenvalue.

To measure the goodness of fit for each point, we consider the errors from projecting each neighboring point onto the fitted plane, which are denoted by \( e^k_i \). Here \( i \) indexes the point corresponding to the fitted plane and \( k \) indexes the neighboring points. The reduced chi-square variable for each point is then given by

\[
\chi^2_i = \frac{1}{\sigma^2 \phi} \sum_k (e^k_i)^2
\]  
(4.2)

where \( \sigma^2 \) is the tolerance of a plane, which is a parameter specified by the user, and \( \phi \) is number of degrees of freedom. The number of degrees of freedom is given by number of neighboring points less the dimensionality of the coordinate space

\[
\phi = |N(x_i)| - 3.
\]  
(4.3)
Once the reduced chi-square variable has been determined for each point, the point is labeled as foliage if $\chi_i^2 \leq 1$ (poor plane fit) or non-foliage if $\chi_i^2 > 1$ (good plane fit). This is technically a detection of non-foliage points since it identifies planes, but we refer to the process as foliage detection for simplicity.

This formulation requires two user selected parameters: $r$, which determines the size of the neighborhood for plane fitting, and $\sigma^2$, which specifies the variance that is expected for a true plane. We essentially chose these values using trial and error and achieved adequate results. However, a selection of parameters based on cross-validation would be more rigorous. A few examples of foliage detection based on the parameters are shown in Figure 4-3. Looking closely at the images shows that a larger variance value yields fewer foliage points. This is expected since some foliage points that are more planar are considered acceptable under a variance of 0.1 meters$^2$. The variance value of 0.05 meters$^2$, however, is more restrictive and accordingly fewer planes are detected. It is clear that a larger $r$ value makes the foliage regions more contiguous. This is particularly apparent with the trees in the upper right quadrant of the image. We ultimately chose values of $r = 2$ meters and $\sigma^2 = 0.05$ meters$^2$ with the intention of detecting foliage and erring on the side of detecting too many foliage points.

Since surface normals are derived as part of the detection process, one could employ the normal vectors as attributes for measuring mutual information. An analysis of this attribution would be interesting, but we did not pursue this topic.

A full characterization of false alarms and missed detections is desirable for evaluating the foliage detection, but it is not practical since it would require manually labeling millions of points. We instead analyze the results qualitatively. Examples of the foliage detection are shown in Figure 4-1 and Figure 4-2. Figure 4-1 shows excellent detection of foliage with barely any noticeable false alarms or missed detections in comparison to the optical image. Figure 4-2 demonstrates a few problems. In addition to detecting foliage, some building edges as well as parked cars are classified as foliage.
Figure 4-1: Example of foliage detection performance. Detected points are shown in green while the non-foliage points are show with elevation-encoded grayscale values. Blue pixels indicate background regions.

4.2 Analysis

The most basic analysis to perform using the foliage segmentation is to look at the joint elevation-luminance characteristics conditioned on whether or not the points are foliage points. This is shown in Figure 4-5 and Figure 4-6. Using the same scene from the previous chapter, we calculated the joint PMF conditioned on non-foliage points (shown in Figure 4-5 (a)) and foliage points (shown in Figure 4-6 (a)). The product of marginal PMFs for non-foliage and foliage points are shown for comparison. Figure 4-4 shows which regions were detected as foliage. It is immediately apparent that the joint characteristics are much different depending on the foliage label. The mutual information for the non-foliage regions is 0.4433 bits (with a maximum MI of 4.4747 bits) while the MI for the foliage regions is 0.3781 bits (with a maximum MI of 6.1567 bits). It is interesting to note that in comparing Figure 4-6 (a) and 4-5 (a) to Figure 3-3 (a), it is apparent that the latter is a combination of PMFs that we have effectively decoupled with the foliage detection.

As with our analysis in the previous chapter, it is necessary to view how the objective function varies as the camera parameters are perturbed before we directly use
Figure 4-2: Example of foliage detection performance. Detected points are shown in green while the non-foliage points are show with elevation-encoded grayscale values. Blue pixels indicate background regions. The square shows a region with false alarms due to building edges; the circle shows false alarms resulting from vehicles.
Figure 4-3: Foliage detection results for different parameters of $r$ and $\sigma^2$. The units are meters and meters$^2$, respectively. The LIDAR elevation and pdet images are shown for comparison.
Figure 4-4: Optical image (a) and LIDAR image (b) showing which points were classified as foliage for the PMFs in Figure 4-5 and Figure 4-6.
Figure 4-5: Joint elevation-luminance PMF and product of marginals PMF for non-foliage regions. The image used for calculating the PMFs and the classification of points is shown in Figure 4-4.
Figure 4-6: Joint elevation-luminance PMF and product of marginals PMF for foliage regions. The image used for calculating the PMFs and the classification of points is shown in Figure 4-4.
it for registration. Another issue specific to this chapter is how to use the foliage segmentation results. Perhaps measuring mutual information with the added binary foliage label produces more reliable results. Or, maybe it is best to completely disregard the foliage points in measuring mutual information and only use the non-foliage points. We answer these questions by looking at a series of probing experiments in the following section.

4.3 Probing Experiments

We perform probing experiments (with the same setup as in the previous chapter) using only foliage points and only non-foliage points. For the foliage points only case, the mutual information is measured between elevation and luminance for the points classified as foliage (and their corresponding optical image pixels) while non-foliage points and background pixels are completely ignored. Similarly, for the non-foliage only case, the mutual information is measured between the elevation and luminance for the points classified as non-foliage while the foliage points and background pixels are completely ignored.

The results of the probing experiments for the eight test images are shown in Figures 4-7, 4-8, 4-9, and 4-10. It is clear that the non-foliage points provide a smoother objective function and a larger basin of attraction. There are a variety of factors that likely contribute to this phenomena. First, foliage, particularly trees, vary in elevation without varying significantly in luminance. The rounded shapes of trees viewed from an aerial perspective contribute to this phenomena. Next, the foliage regions by definition indicate regions with rapidly changing elevation values. While these points may be useful for refining a precise registration, they are not helpful for coarse registrations because they give no reliable indication of which way the parameters should be changed to decrease the objective value. This is similar to the argument given for why matching of edges is not useful for registration: there is a sharp spike in the objective function at the correct parameter values, and the rest of the objective function is useless for search-based optimization.
Figure 4-7: Probing experiments measuring the normalized joint entropy of elevation and luminance for foliage regions and non-foliage regions.
Figure 4-8: Probing experiments measuring the normalized joint entropy of elevation and luminance for foliage regions and non-foliage regions.
Figure 4-9: Probing experiments measuring the normalized joint entropy of elevation and luminance for foliage regions and non-foliage regions.
Figure 4-10: Probing experiments measuring the normalized joint entropy of elevation and luminance for foliage regions and non-foliage regions.
4.4 Registration Tests

The registration experiments that we show compare the performance of the method described in this chapter with the those described in the previous chapter. We apply the same technique of starting with randomly selected perturbations of camera parameters and analyzing how many registration results are correct. The same randomly selected starting locations from the previous chapter are used here for consistency. We perform tests for the small range parameters (shown in Table 3.1) as well as the large range parameters (shown in Table 3.2).

Results for the small range parameters are shown in Tables 4.1, 4.2, and 4.3. The results show a significant improvement in accuracy compared to the elevation-based results from the previous chapter. The accuracy is also comparable to the pdet and dual accuracies described in the previous chapter. This is significant since the foliage detection method requires only 3D coordinates and no pdet values.

The results for the large range parameters are more significant, as shown in Tables 4.4, 4.5, and 4.6. Here, the non-foliage method has an accuracy of 81.0% (compared to 66.5% for the elevation attribution) with roughly the same mean registration time. The non-foliage method outperforms the dual attribution method from the previous chapter (75.6%), which is particularly notable since the non-foliage method does not use pdet values as the dual method does. Overall, the results show that the foliage segmentation technique is useful for registration with a coarse starting approximation and for fine alignment. An example of a random perturbation drawn from range given in Table 3.2 and the post-algorithm registration is shown in Figure 4-15.
Table 4.1: Number of correctly registered images (out of 100) for the small range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter.

<table>
<thead>
<tr>
<th>JE Measure:</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
<th>non-Folg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>100</td>
<td>93</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Image 2</td>
<td>97</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>Image 3</td>
<td>83</td>
<td>100</td>
<td>96</td>
<td>99</td>
</tr>
<tr>
<td>Image 4</td>
<td>76</td>
<td>83</td>
<td>97</td>
<td>95</td>
</tr>
<tr>
<td>Image 5</td>
<td>97</td>
<td>97</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>Image 6</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Image 7</td>
<td>96</td>
<td>95</td>
<td>97</td>
<td>100</td>
</tr>
<tr>
<td>Image 8</td>
<td>99</td>
<td>99</td>
<td>100</td>
<td>99</td>
</tr>
<tr>
<td>Total:</td>
<td>93.5%</td>
<td>95.8%</td>
<td>98.5%</td>
<td>99.9%</td>
</tr>
</tbody>
</table>

Figure 4-11: Bar chart for values given in Table 4.1. The error bars indicate the minimum and maximum values.
Table 4.2: Mean registration times in seconds (for correctly registered images). The times correspond to tests from the small range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter.

<table>
<thead>
<tr>
<th>JE Measure:</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
<th>non-Folg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>6.30</td>
<td>8.12</td>
<td>14.18</td>
<td>6.57</td>
</tr>
<tr>
<td>Image 2</td>
<td>6.80</td>
<td>7.14</td>
<td>13.39</td>
<td>9.09</td>
</tr>
<tr>
<td>Image 3</td>
<td>6.44</td>
<td>8.59</td>
<td>14.67</td>
<td>6.74</td>
</tr>
<tr>
<td>Image 4</td>
<td>5.76</td>
<td>5.79</td>
<td>10.88</td>
<td>6.34</td>
</tr>
<tr>
<td>Image 5</td>
<td>6.55</td>
<td>6.58</td>
<td>12.39</td>
<td>6.62</td>
</tr>
<tr>
<td>Image 6</td>
<td>6.19</td>
<td>7.13</td>
<td>11.89</td>
<td>6.73</td>
</tr>
<tr>
<td>Image 7</td>
<td>6.77</td>
<td>8.27</td>
<td>14.44</td>
<td>6.72</td>
</tr>
<tr>
<td>Image 8</td>
<td>5.99</td>
<td>6.53</td>
<td>12.40</td>
<td>6.48</td>
</tr>
<tr>
<td>Total:</td>
<td>6.35</td>
<td>7.27</td>
<td>13.03</td>
<td>6.91</td>
</tr>
</tbody>
</table>

Figure 4-12: Bar chart for values given in Table 4.2. The error bars indicate standard deviations.
Table 4.3: Mean number of iterations (for correctly registered images). The times correspond to tests from the small range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter.

<table>
<thead>
<tr>
<th>JE Measure:</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
<th>non-Folg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>114.8</td>
<td>115.9</td>
<td>112.1</td>
<td>118.7</td>
</tr>
<tr>
<td>Image 2</td>
<td>127.4</td>
<td>123.1</td>
<td>118.5</td>
<td>128.5</td>
</tr>
<tr>
<td>Image 3</td>
<td>116.7</td>
<td>119.9</td>
<td>115.0</td>
<td>124.7</td>
</tr>
<tr>
<td>Image 4</td>
<td>98.5</td>
<td>104.0</td>
<td>99.6</td>
<td>116.5</td>
</tr>
<tr>
<td>Image 5</td>
<td>112.4</td>
<td>116.6</td>
<td>112.9</td>
<td>123.1</td>
</tr>
<tr>
<td>Image 6</td>
<td>114.9</td>
<td>116.9</td>
<td>110.0</td>
<td>121.5</td>
</tr>
<tr>
<td>Image 7</td>
<td>118.4</td>
<td>115.2</td>
<td>110.5</td>
<td>123.4</td>
</tr>
<tr>
<td>Image 8</td>
<td>111.2</td>
<td>118.4</td>
<td>108.8</td>
<td>120.4</td>
</tr>
<tr>
<td>Total:</td>
<td>115.6</td>
<td>116.2</td>
<td>110.9</td>
<td>122.1</td>
</tr>
<tr>
<td>JE Measure:</td>
<td>Elev</td>
<td>Pdet</td>
<td>Dual</td>
<td>non-Folg</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>Image 1</td>
<td>91</td>
<td>61</td>
<td>84</td>
<td>85</td>
</tr>
<tr>
<td>Image 2</td>
<td>49</td>
<td>78</td>
<td>74</td>
<td>83</td>
</tr>
<tr>
<td>Image 3</td>
<td>52</td>
<td>90</td>
<td>67</td>
<td>71</td>
</tr>
<tr>
<td>Image 4</td>
<td>31</td>
<td>29</td>
<td>52</td>
<td>49</td>
</tr>
<tr>
<td>Image 5</td>
<td>84</td>
<td>69</td>
<td>81</td>
<td>87</td>
</tr>
<tr>
<td>Image 6</td>
<td>98</td>
<td>87</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>Image 7</td>
<td>43</td>
<td>52</td>
<td>63</td>
<td>86</td>
</tr>
<tr>
<td>Image 8</td>
<td>84</td>
<td>60</td>
<td>85</td>
<td>87</td>
</tr>
<tr>
<td>Total</td>
<td>66.5%</td>
<td>65.8%</td>
<td>75.6%</td>
<td>81.0%</td>
</tr>
</tbody>
</table>

Table 4.4: Number of correctly registered images (out of 100) for the large range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter.

Figure 4-13: Bar chart for values given in Table 4.4. The error bars indicate the minimum and maximum values.
Table 4.5: Mean registration times in seconds (for correctly registered images). The times correspond to tests from the large range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter.

<table>
<thead>
<tr>
<th>JE Measure: Elev Pdet Dual non-Folg</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
<th>non-Folg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>8.72</td>
<td>10.65</td>
<td>29.95</td>
<td>9.57</td>
</tr>
<tr>
<td>Image 2</td>
<td>8.21</td>
<td>9.76</td>
<td>29.46</td>
<td>9.14</td>
</tr>
<tr>
<td>Image 3</td>
<td>9.92</td>
<td>10.29</td>
<td>28.65</td>
<td>8.93</td>
</tr>
<tr>
<td>Image 4</td>
<td>7.23</td>
<td>8.47</td>
<td>24.56</td>
<td>7.49</td>
</tr>
<tr>
<td>Image 5</td>
<td>8.86</td>
<td>8.60</td>
<td>25.61</td>
<td>8.26</td>
</tr>
<tr>
<td>Image 6</td>
<td>8.07</td>
<td>11.42</td>
<td>26.42</td>
<td>8.21</td>
</tr>
<tr>
<td>Image 7</td>
<td>9.93</td>
<td>9.26</td>
<td>29.19</td>
<td>9.19</td>
</tr>
<tr>
<td>Image 8</td>
<td>7.91</td>
<td>7.66</td>
<td>25.65</td>
<td>10.83</td>
</tr>
<tr>
<td>Total</td>
<td>8.61</td>
<td>9.51</td>
<td>27.44</td>
<td>8.95</td>
</tr>
</tbody>
</table>

Figure 4-14: Bar chart for values given in Table 4.5. The error bars indicate standard deviations.
<table>
<thead>
<tr>
<th>JE Measure:</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
<th>non-Folg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>161.4</td>
<td>153.1</td>
<td>149.8</td>
<td>164.7</td>
</tr>
<tr>
<td>Image 2</td>
<td>143.4</td>
<td>150.2</td>
<td>149.1</td>
<td>169.3</td>
</tr>
<tr>
<td>Image 3</td>
<td>144.4</td>
<td>144.6</td>
<td>138.4</td>
<td>155.4</td>
</tr>
<tr>
<td>Image 4</td>
<td>126.4</td>
<td>116.5</td>
<td>120.3</td>
<td>139.8</td>
</tr>
<tr>
<td>Image 5</td>
<td>148.3</td>
<td>142.0</td>
<td>141.2</td>
<td>155.2</td>
</tr>
<tr>
<td>Image 6</td>
<td>150.7</td>
<td>147.1</td>
<td>139.9</td>
<td>153.2</td>
</tr>
<tr>
<td>Image 7</td>
<td>143.3</td>
<td>128.6</td>
<td>140.3</td>
<td>166.6</td>
</tr>
<tr>
<td>Image 8</td>
<td>145.7</td>
<td>141.1</td>
<td>137.8</td>
<td>153.3</td>
</tr>
<tr>
<td>Total</td>
<td>145.5</td>
<td>140.4</td>
<td>139.6</td>
<td>157.2</td>
</tr>
</tbody>
</table>

Table 4.6: Mean number of iterations (for correctly registered images). The times correspond to tests from the large range parameters. The non-foliage method results are shown in comparison to methods from the previous chapter.
Figure 4-15: Example of registration using an initial perturbation from Table 3.2. Registration was performed by measuring the joint entropy between luminance and elevation for non-foliage regions.
Chapter 5

Multi-View Registration

In many scenarios, LIDAR data is collected infrequently, while optical images of a given scene are collected repeatedly over time and from a variety of perspectives. It is thus useful to consider methods for incorporating previously registered images to register a newly acquired image. Even if one is interested in merely determining the location of one optical image relative to another optical image, the LIDAR data can be beneficial. When LIDAR data is not available, it is often difficult to align two 2D images of the same scene taken from drastically different perspectives. This is due to parallax issues resulting from elevated structures. When LIDAR data is available, however, a single image can be used to create a 3D virtual reality model, providing structure and occlusion reasoning for subsequent registrations. This chapter provides a quantitative analysis of the utility provided by an existing registered image for registering a new image. We describe a simple method for fusing optical and LIDAR imagery and implementing it in a registration algorithm. We present two attribution methods that make use of the registered image, and compare their performance with the other methods from previous chapters.

5.1 3D Model Generation

Image based rendering is a popular method for generating 3D virtual reality models [4, 23]. While such methods vary widely in level of sophistication, they all ultimately
involve using one or more images, often with a geometric model, to render scenes from new perspectives. We describe our implementation in two sections: structure inference and projective texture mapping.

5.1.1 Structure Inference

The first issue relevant to 3D model generation is inferring a surface from the 3D point cloud. There are certainly many ways to approach this task. For example, a method combining information from the LIDAR point cloud and the registered optical image could be used to infer surfaces, and even refine the geometry provided by the LIDAR data. This approach would likely produce precise results, but would be computationally expensive. Since we are interested in a relatively fast and simple method for inferring structure, we use a simple technique based on Delaunay triangulation [3].

Two features of our data make it amenable to a Delaunay triangulation for surface inference. First, the LIDAR data was taken from a near nadir position, so there is typically only one z value for a given (x,y) coordinate (occasional exceptions arise from overlapping swaths, which are inconsequential). Next, the LIDAR data has a fairly dense sampling resolution – about 5 points per square meter. Since the sampling of points is very high relative to the elevation variation of the scene, the Delaunay triangulation provides a close approximate to the true underlying 3D structure.

Figure 5-1 shows the details of how we implement the Delaunay triangulation. We only use the (x,y) coordinates in the triangulation process to determine the point connectivity. This is necessary for connecting points with a steep slope, such as those that correspond to building walls. The connectivity is then used to draw triangles in 3D space. A few properties of the Delaunay triangulation are worth noting. The algorithm seeks to maximize the minimum angle of all the angles of the triangles in the triangulation. The triangulation also has the property that a circle circumscribed on any triangle does not contain other points. An example of a 3D mesh and the corresponding point cloud is shown in Figure 5-2. The mesh has a jagged appearance, but this effect is mitigated when the mesh is textured with an image.
Figure 5-1: Depiction of Delaunay triangulation process for inferring a 3D surface from LIDAR point cloud.
5.1.2 Projective Texture Mapping

The second major step in 3D model generation is texture mapping a registered image onto the mesh. As with structure inference, there are many sophisticated methods for approaching this problem. Ideally, one would attribute a texture unique to each mesh element. This would work especially well if multiple registered images existed, as different mesh elements could be used to texture different triangles. The selection of textures for each mesh element could be performed automatically by choosing texture images that have a direct and high-resolution view of the element. However, this requires a complicated implementation and yields longer rendering times. Since we only consider the case with one given registered image, we use the automatic texture coordinate generation facility in OpenGL, in conjunction with shadow mapping techniques for occlusion reasoning [7, 8].

The texture mapped scene is rendered with a series of three renderings in OpenGL. We refer to the OpenGL camera parameters for the image to be texture mapped as the projector and the camera parameters for the camera describing the view of the user as the view camera. The mesh is first rendered from the perspective of the projector to obtain a depth map, where different intensities indicate different distances from the projector. The second rendering is from the perspective of the view camera, and the optical image is textured onto the scene using four dimensional automatic texture generation in OpenGL [7]. At this point, there has been no occlusion reasoning. The third rendering is also from the perspective of the view camera and utilizes the depth map from the first rendering as a mask. In regions that were occluded from the perspective of the projector, the scene is colored black (or blue to indicate an occluded region during registration).

An example of a texture mapped mesh is shown in Figure 5-2 (c). This demonstrates that the mesh looks much smoother after it has been texture mapped, which is a well known phenomena in the computer graphics community [4, 23]. The blue regions in the model indicate areas that were obstructed when the aerial photograph was taken.
Figure 5-2: Demonstration of texture mapping showing LIDAR point cloud (a), Delaunay mesh (b), and texture mapped mesh (c). The blue regions in (c) indicate areas where the mesh could not be textured due to occlusion.
5.2 Attribution Methods

There are a variety of ways to incorporate the additional information provided by the given registered image. We present two methods here, which are by no means an exhaustive evaluation of potential methods. We first present the most basic method for using the registered image, which uses only luminance from the registered image and luminance from the image. The second method incorporates luminance from both images, as well as the non-foliage attribute from the previous chapter.

5.2.1 Luminance

With an existing registered image, it makes sense to first analyze how to exclusively use both images, as they are of the same mode. This attribution uses the LIDAR data only as a source of geometry for texture mapping the registered image. To avoid confusion, we refer to the existing registered image as the registered image and the image to be registered as the image. The features attributed to the LIDAR data are the luminance values from the registered image. Luminance is also used for the image. Using luminance instead of color allows for a simple reduction in the necessary number of histogram bins from $2^{48}$ to $2^{16}$. This is a somewhat arbitrary discretization, but it yields a reasonable ratio of samples to histogram bins while maintaining the appearance of the scene. We briefly experimented with slightly different numbers of discretization levels and found that this one worked best. One could consider the use of correlation for an objective function here instead of mutual information since images of the same mode are being registered. However, changes in shadows, illumination, and scene content could introduce nonlinear errors, which are often handled much better by mutual information than correlation.

To reduce the number of mesh triangles that must be rendered, we randomly sample 10% of the points from the LIDAR files and perform the Delaunay triangulation with this subset of points. We found that this did not significantly affect the appearance of the texture mapped models, while it greatly reduced the rendering times in OpenGL. Only regions in the 3D mesh that are texture mapped with the
Figure 5-3: Screenshot from multi registration algorithm. Blue pixels indicate occluded regions or background regions. These pixels are not used in the registration algorithm.

existing registered image are used in registration. This means that regions that are not texture mapped due to occlusion, as shown in Figure 5-2, are not used. Figure 5-3 shows a screenshot from the registration algorithm, where the blue pixels indicate occluded regions or background regions.

5.2.2 Non-foliage and Luminance

While the previous attribution is simple and intuitive, it does not explicitly utilize the properties of the LIDAR such as elevation. Here, we explain one way of using the LIDAR elevation in addition to LIDAR luminance and image luminance. By LIDAR luminance, we mean the luminance from the registered image that has been mapped onto the LIDAR data. Similar to the dual attribution method that we describe in
Chapter 3, we are interested in measuring

\[ I(v_e, v_l; u), \]

where \( u \) is the image luminance, \( v_e \) is the LIDAR elevation, and \( v_l \) is the LIDAR luminance. This can be expressed in terms of joint and marginal entropies:

\[ I(v_e, v_p; u) = H(v_e, v_p) + H(u) - H(u, v_e, v_p). \]  

(5.1)

We can again employ an independence assumption to simplify calculating the entropy term with three variables. We assume that the LIDAR elevation is independent of the LIDAR luminance conditioned on the image luminance (for approximately registered images). An interpretation of this statement is as follows: say that we are given the image luminance for a given pixel and we want to predict the LIDAR luminance. Knowledge of the LIDAR elevation for this pixel does not provide any extra information for estimating the LIDAR luminance. This intuitively makes sense, and is likely more justified than the assumption in Chapter 3, because there is a more direct relationship between LIDAR luminance and image luminance than LIDAR pdet and image luminance. The independence assumption leads to the following approximation

\[ H(u, v_e, v_l) = H(u, v_l) + H(u, v_e) - H(u). \]  

(5.2)

Since the analysis in Chapter 4 shows that it is useful to ignore foliage regions, we employ the same approach here when dealing with LIDAR elevation. Yet, there is no reason to ignore foliage regions in the LIDAR luminance, as they are likely to match with image luminance. This leads to a slightly different objective function

\[ H(u, v_e, v_l) \approx H(u, v_l) + H(u, v^*_e) - H(u), \]  

(5.3)

where \( v^*_e \) is the LIDAR elevation for non-foliage regions.

This formulation is somewhat complicated, so we briefly describe how the algo-
algorithm is implemented. Each measurement of joint entropy requires two renderings. The first rendering is the LIDAR elevation with special labels indicating foliage points. \( H(u, v^*_f) \) is calculated using only non-foliage and non-background points and their corresponding optical image pixels (determined via projection). The second rendering is of the Delaunay mesh with the registered image texture mapped onto the mesh. This rendering looks like Figure 5-3. No distinction is made between foliage and non-foliage points for this rendering. \( H(u, v_l) \) is calculated using non-background pixels and their corresponding optical image pixels. Finally, \( H(u, v^*_f) \) is added to \( H(u, v_l) \) to achieve the final objective value. Note that the inclusion of \( H(u) \) is not necessary in the optimization procedure because it remains constant. We provide a more extensive description of this algorithm and implementation in the appendix.

5.3 Probing Experiments

We performed probing experiments using the same technique described in the last two chapters. The probing experiments are shown in Figures 5-4, 5-5, 5-6, and 5-7. The plots show the normalized objective function for the luminance attribution, which we refer to as the “multi” method, the non-foliage and luminance attribution, which we refer to as the “multi-non-folg” method, and the non-foliage elevation-luminance measure described in the previous chapter, which we refer to as the “non-folg” method.

The plots for the multi and multi-non-folg measures (which we refer to as the multi-view measures) demonstrate convexity and smoothness properties comparable to those of the non-foliage method. For essentially all images, the objective functions are narrower for the multi-view attributions; it is unclear if this affects registration performance. The plots for image 4 show the most drastic difference in performance, where the multi-view functions are quasiconvex over a larger parameter range. This is easily explained by the structure of the image: the scene has little elevation variation in comparison to the luminance variation. Thus the multi-view methods are more
desirable in this case. Overall, the probing experiments predict that the registration performance of the multi-view attributions will be similar to the performance of the non-foliage method.

5.4 Registration Tests

As with previous registration tests, we tested the multi-view registration algorithms with randomly sampled perturbations. We used the same starting perturbations that were used in the previous chapters for consistency. In contrast to our other registration methods, the post-algorithm registrations were not clearly correct or incorrect. There were fewer optimization runaways, and many registrations were close to the correct alignment, but not sufficient for projective texture mapping. For consistency, we used the same criteria for a correct registration that was used in the previous chapters. The correspondence point mean squared error (MSE) was measured for all post-algorithm registrations. Registrations with a MSE less than 50 pixels$^2$ were considered correct. Again this is sufficiently precise for performing projective texture mapping with negligible errors. Since the methods discussed in previous chapters demonstrate nearly perfect performance for the small range parameter perturbations, we only used the large range perturbation parameters (listed in Table 3.2) for the registration tests.

The results are shown in in Tables 5.1, 5.2, and 5.3. Again, we refer to the first attribution method in this chapter as the "multi" attribution and the second method as the "multi-non-folg" attribution. The accuracy for the multi attribution is 82.6%, which is significantly higher than the elevation and pdet attributions. This is expected since the algorithm directly measures MI between the luminance of two images, rather than the luminance and pdet for example, which is a multi-modal situation. This is best understood by considering that high mutual information corresponds to the ability to predict a value in one image given a value from the other. It is easier to predict a luminance value given another luminance value than to predict a luminance value given an elevation value, for example. The multi method does not, however, perform significantly better than the non-folg method.
Figure 5-4: Probing experiments for Image 1 and Image 2 measuring the normalized joint entropy with the two attributions described in this chapter and the non-foliage method from the previous chapter.
Figure 5-5: Probing experiments for Image 3 and Image 4 measuring the normalized joint entropy with the two attributions described in this chapter and the non-foliage method from the previous chapter.
Figure 5-6: Probing experiments for Image 5 and Image 6 measuring the normalized joint entropy with the two attributions described in this chapter and the non-foliage method from the previous chapter.
Figure 5-7: Probing experiments for Image 7 and Image 8 measuring the normalized joint entropy with the two attributions described in this chapter and the non-foliage method from the previous chapter.
It is worthwhile to look at the particular images with the largest discrepancies in accuracy between the non-foliage method and the multi method. The most obvious difference is for Image 4, where the multi method performs significantly better (76%) than the non-Foliage method (49%). As described in the probing experiments section, this can be explained by the structure of the scene. Image 4 has very little change in elevation as compared to other images. The luminance values for the scene are comparatively more detailed, which makes the multi attribution a preferable choice for this particular scene. In contrast, the non-foliage method performs much better for Image 8. Image 8 has a fair amount of variation in elevation, particularly due to the large building at the bottom of the image.

The best results are given by the multi-non-folg algorithm, which shows an accuracy significantly higher than the non-folg method (91.1% versus 81.0%). In contrast to the multi method, this attribution clearly shows the marginal utility provided by the extra registered image. This is because the multi-non-folg method explicitly uses elevation and the luminance from the existing registered image to perform registration. This verifies our intuitive knowledge about information and the scene: that more information (given by the existing registered image) leads to a more accurate estimation of camera parameters. It is important to note that introducing even more information (e.g. additional registered images) would likely provide a more improved estimate, but there are decreasing marginal returns in utility provided by each subsequent image. Ultimately, 100 registered images would probably not provide better registration than 50 registered images, for example. The main drawback the algorithm is the longer mean registration time, which is 46.34 seconds. This is expected due to the need for multiple renderings. This attribution method represents the final progressive step in our development, bringing the overall accuracy from the initial 66.5% to 91.1% for the large range parameter perturbations.
Table 5.1: Number of correctly registered images out of 100 randomly sampled perturbations for multi and multi-non-folg attributions. Attributions from previous chapters are also shown. The perturbations were sampled from the large parameter range listed in Table 3.2.

<table>
<thead>
<tr>
<th>Image</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
<th>non-Folg</th>
<th>Multi</th>
<th>Multi-non-Folg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>91</td>
<td>61</td>
<td>84</td>
<td>85</td>
<td>90</td>
<td>95</td>
</tr>
<tr>
<td>Image 2</td>
<td>49</td>
<td>78</td>
<td>74</td>
<td>83</td>
<td>66</td>
<td>78</td>
</tr>
<tr>
<td>Image 3</td>
<td>52</td>
<td>90</td>
<td>67</td>
<td>71</td>
<td>88</td>
<td>85</td>
</tr>
<tr>
<td>Image 4</td>
<td>31</td>
<td>29</td>
<td>52</td>
<td>49</td>
<td>76</td>
<td>86</td>
</tr>
<tr>
<td>Image 5</td>
<td>84</td>
<td>69</td>
<td>81</td>
<td>87</td>
<td>91</td>
<td>95</td>
</tr>
<tr>
<td>Image 6</td>
<td>98</td>
<td>87</td>
<td>99</td>
<td>100</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>Image 7</td>
<td>43</td>
<td>52</td>
<td>63</td>
<td>86</td>
<td>90</td>
<td>91</td>
</tr>
<tr>
<td>Image 8</td>
<td>84</td>
<td>60</td>
<td>85</td>
<td>87</td>
<td>61</td>
<td>99</td>
</tr>
<tr>
<td>Total</td>
<td>66.5%</td>
<td>65.8%</td>
<td>75.6%</td>
<td>81.0%</td>
<td>82.6%</td>
<td>91.1%</td>
</tr>
</tbody>
</table>

Figure 5-8: Bar chart for values given in Table 5.1. The error bars indicate the minimum and maximum values.
Table 5.2: Mean registration times in seconds for multi and multi-non-folg attributions. Attributions from previous chapters are also shown. Each image was tested with 100 randomly chosen perturbations. The perturbations were sampled from the large parameter range listed in Table 3.2.

<table>
<thead>
<tr>
<th>Image</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
<th>non-Folg</th>
<th>Multi</th>
<th>Multi-non-Folg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>8.72</td>
<td>10.65</td>
<td>29.95</td>
<td>9.57</td>
<td>14.00</td>
<td>49.91</td>
</tr>
<tr>
<td>Image 4</td>
<td>7.23</td>
<td>8.47</td>
<td>24.56</td>
<td>7.49</td>
<td>10.96</td>
<td>52.47</td>
</tr>
<tr>
<td>Image 5</td>
<td>8.86</td>
<td>8.60</td>
<td>25.61</td>
<td>8.26</td>
<td>8.67</td>
<td>37.14</td>
</tr>
<tr>
<td>Image 6</td>
<td>8.07</td>
<td>11.42</td>
<td>26.42</td>
<td>8.21</td>
<td>9.53</td>
<td>38.49</td>
</tr>
<tr>
<td>Image 7</td>
<td>9.93</td>
<td>9.26</td>
<td>29.19</td>
<td>9.19</td>
<td>13.11</td>
<td>46.16</td>
</tr>
<tr>
<td>Image 8</td>
<td>7.91</td>
<td>7.66</td>
<td>25.65</td>
<td>10.83</td>
<td>10.29</td>
<td>43.56</td>
</tr>
<tr>
<td>Total</td>
<td>8.61</td>
<td>9.51</td>
<td>27.44</td>
<td>8.95</td>
<td>11.87</td>
<td>46.34</td>
</tr>
</tbody>
</table>

Figure 5-9: Bar chart for values given in Table 5.2. The error bars indicate standard deviations.
<table>
<thead>
<tr>
<th>Image</th>
<th>Elev</th>
<th>Pdet</th>
<th>Dual</th>
<th>non-Folg</th>
<th>Multi</th>
<th>Multi-non-Folg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>161.4</td>
<td>153.1</td>
<td>149.8</td>
<td>164.7</td>
<td>200.4</td>
<td>188.1</td>
</tr>
<tr>
<td>Image 2</td>
<td>143.4</td>
<td>150.2</td>
<td>149.1</td>
<td>169.3</td>
<td>216.9</td>
<td>200.2</td>
</tr>
<tr>
<td>Image 3</td>
<td>144.4</td>
<td>144.6</td>
<td>138.4</td>
<td>155.4</td>
<td>204.0</td>
<td>184.0</td>
</tr>
<tr>
<td>Image 4</td>
<td>126.4</td>
<td>116.5</td>
<td>120.3</td>
<td>139.8</td>
<td>207.8</td>
<td>199.1</td>
</tr>
<tr>
<td>Image 5</td>
<td>148.3</td>
<td>142.0</td>
<td>141.2</td>
<td>155.2</td>
<td>151.9</td>
<td>148.0</td>
</tr>
<tr>
<td>Image 6</td>
<td>150.7</td>
<td>147.1</td>
<td>139.9</td>
<td>153.2</td>
<td>165.2</td>
<td>151.2</td>
</tr>
<tr>
<td>Image 7</td>
<td>143.3</td>
<td>128.6</td>
<td>140.3</td>
<td>166.6</td>
<td>186.0</td>
<td>174.1</td>
</tr>
<tr>
<td>Image 8</td>
<td>145.7</td>
<td>141.1</td>
<td>137.8</td>
<td>153.3</td>
<td>194.8</td>
<td>169.5</td>
</tr>
<tr>
<td>Total</td>
<td>145.5</td>
<td>140.4</td>
<td>139.6</td>
<td>157.2</td>
<td>190.9</td>
<td>176.8</td>
</tr>
</tbody>
</table>

Table 5.3: Mean number of iterations for multi and multi-non-folg attributions. Attributions from previous chapters are also shown. Each image was tested with 100 randomly chosen perturbations. The perturbations were sampled from the large parameter range listed in Table 3.2.
Chapter 6

Conclusion

In this thesis, we have presented a series of novel algorithms for performing automatic registration of optical and LIDAR imagery. The algorithms are fast compared to previous registration methods, with runtimes on the order of seconds. This was accomplished by employing OpenGL and advanced graphics hardware to perform registration. All of the algorithms work with ungridded LIDAR data, and many of them do not require LIDAR probability of detection (pdet) values. The basic algorithms exploit the statistical dependency between optical luminance and LIDAR elevation that is present in urban scenes. We improved the basic algorithms with a novel application of foliage detection, where the exclusion of points declared as foliage (using the technique presented in [28]) improves registration reliability significantly. Finally, we showed how the use of an existing registered optical image can be used to achieve even more reliable registration.

We began with an analysis of the statistical dependencies existing among LIDAR elevation, LIDAR probability of detection (pdet), and image luminance. These dependencies were validated empirically with mutual information measures, where MI measures are higher for registered images than non-registered images. This was followed with a series of probing experiments showing that the objective function is relatively smooth and quasiconvex for the eight images of urban scenes. We then showed registration experiments where we simulated initial starting locations and analyzed the post-algorithm accuracy. For the small range initial starting positions,
the accuracy rates were 93.5%, 95.8%, and 98.5% for the elevation, pdet, and dual methods, respectively. Similarly, the accuracy rates for the large range parameters were 66.5%, 65.8%, and 75.6%. All registration times averaged less than 30 seconds.

Building on these basic algorithms, we demonstrated how LIDAR foliage segmentation can improve registration accuracy. To separate foliage and non-foliage points, we used a planar goodness of fit test with the neighborhood of points around each point, which was originally suggested by [28]. We showed that while both foliage points and non-foliage points show measurable dependence, using only the non-foliage points yields a much smoother objective function. This was verified with registration tests. The improved accuracy rate for the large range parameters was 81.0%. The result is significant since the algorithm does not require pdet values. Registration runtimes averaged less than 10 seconds with this algorithm.

Finally, we introduced a method for utilizing an existing registered image to improve registration accuracy. We described two attribution methods: one simply using the luminance from both images (multi), and another using the luminance from both images as well as LIDAR elevation with the exclusion of foliage points (multi-non-folg). The second method utilizes a conditional independence term to simplify calculations. The registration accuracies corresponding to the large range parameters were 82.6% and 91.1% for the multi and multi-non-folg methods, respectively. The multi-non-folg method demonstrates the best performance of all the algorithms in the thesis, but has a notably larger average runtime of 46.34 seconds.

6.1 Future Work

An important extension to the work presented here is testing the algorithms on more and different types of data. The LIDAR data that we used has a relatively high sampling density, so it would be useful to test the algorithms on other datasets that are sparser. Such datasets could be simulated by using a subset of the points from the LIDAR data that we used. Even more important is testing the algorithms with different types of optical imagery, taken from both aerial and ground perspectives.
The optical images that we used were taken from an oblique aerial perspective and at
only one geographical region. While we anticipate that the algorithms would perform
well with nadir imagery, this needs to be tested. The algorithms also need to be tested
on dense urban areas, as well as barren scenes that have few man-made objects. It
is also worth looking at optical images taken during different seasons, as this has a
dramatic effect on the appearance of foliage.

We assumed a coarse initial estimate of camera parameters in our registration
problem. While this approach is conventional, it would be interesting to study tech-
niques that would not require this initial estimate. Ideally, one could develop a
registration technique that would only require knowledge of optical image placement
within a few thousand meters relative to the LIDAR data. This would likely require
the use of different features, and possibly the use of different registration approaches,
such as RANSAC [9].

Similarly, the use of multi-resolution registration techniques might prove useful.
Since we have shown how precise camera parameters must be to achieve reliable
registration with our algorithms, one could develop a method that starts with less
accurate initial camera parameters and refines the parameters to the precision of our
algorithms. Our algorithms could then perform the fine registration. We briefly stud-
ied simple multi-resolution techniques using the attributes presented in the thesis with
less precise camera parameters, but found that it did not provide a significant benefit.
There are however, many other ways of employing multi-resolution techniques, such
as using different feature attributions or using model-based registration methods.

Our use of foliage detection is a primitive approach to answering the following
question: which image regions are most useful for registration? There are many
approaches for pursuing this problem. Building on the foliage detection technique,
it would be interesting to experiment with different levels of point neighborhood
variance rather than using the binary label of foliage or non-foliage. Such levels could
be weighted according to their usefulness for registration. Furthermore, one could
use different detection techniques for identifying regions useful to registration, both
in the optical image and LIDAR data.
To expand the applicability of our algorithms, it would be interesting to experiment with additional image modes, such as synthetic aperture radar or infrared imagery. Mutual information is well suited for multi modal image registration, so our algorithms could be used directly for new image modes, assuming that the appropriate LIDAR data is available. It would be particularly useful to test the multi-view algorithm with other image modes, since there are likely scenarios where one image mode is finely registered with the LIDAR data while the other mode has only a coarse registration.

Finally, an important aspect of the multi-view algorithms is the method for obtaining 3D structure from the LIDAR point cloud. The Delaunay triangulation gives a jagged representation of structure, and is not optimal for draping foliage regions. Methods for refining and simplifying the mesh would likely yield better registration results and possibly decreased runtimes [11].
Appendix A

Proofs

A.1 Joint Entropy Approximation

In our development of the Dual attribution method in Chapter 2, we make the following assumption regarding the physical measurements of luminance, elevation, and pdet when the data are registered:

\[ \arg\min_T H(u, v_e, v_p; T) = \arg\min_T [H(u, v_e; T) + H(u, v_p; T)]. \]  \( \text{(A.1)} \)

We wish to show that this statement is true under the assumption that pdet is statistically independent of elevation conditioned on the optical image luminance values. It is important to note that this assumption is only an approximation, and that it does not hold when the data are not registered. For sake of simplicity, we adopt a notation where \( D \) represents the LIDAR probability of detection (pdet) values, \( E \) stands for the LIDAR elevation values, and \( L \) is the optical image luminance values. With the change of notation, we wish to prove

\[ \arg\min_T H(D, E, L) = \arg\min_T [H(D, L) + H(E, L)]. \]  \( \text{(A.2)} \)
Starting with the chain rule for joint entropy, we have

\[ H(D, E, L) = H(D, E|L) + H(L). \]  
(A.3)

We now focus on the conditional term,

\[ H(D, E|L) = \sum_{l \in L} p(l) H(D, E|L = l) \]  
(A.4)

which is given by the definition of conditional entropy. Further manipulation of this term yields

\[ H(D, E|L) = \sum_{l \in L} p(l) \sum_{d \in D} \sum_{e \in E} p(d, e|l) \log p(d, e|l). \]  
(A.5)

Assuming independence of the pdet and elevation values conditioned on the luminance values, we have

\[ p(D, E|L) = p(D|L)p(E|L). \]  
(A.6)

This leads to

\[ H(D, E|L) = \sum_{l \in L} p(l) \sum_{d \in D} \sum_{e \in E} p(d|l)p(e|l) \log p(d|l) + \log p(e|l) \]  
(A.7)

\[ = \sum_{l \in L} p(l) \sum_{d \in D} \sum_{e \in E} p(d|l)p(e|l) \log p(d|l) + \sum_{l \in L} p(l) \sum_{d \in D} \sum_{e \in E} p(d|l)p(e|l) \log p(e|l) \]  
(A.8)

\[ = \sum_{l \in L} p(l) \sum_{d \in D} p(d|l) \log p(d|l) \sum_{e \in E} p(e|l) + \sum_{l \in L} p(l) \sum_{e \in E} p(e|l) \log p(e|l) \sum_{d \in D} p(d|l) \]  
(A.9)

\[ = \sum_{l \in L} p(l) \sum_{d \in D} p(d|l) \log p(d|l) + \sum_{l \in L} p(l) \sum_{e \in E} p(e|l) \log p(e|l) \]  
(A.10)

\[ = \sum_{l \in L} p(l) H(D|L = l) + \sum_{l \in L} p(l) H(E|L = l) \]  
(A.11)

\[ = H(D|L) + H(E|L). \]  
(A.12)

We now have

\[ H(D, E, L) = H(D|L) + H(E|L) + H(L). \]  
(A.13)
Since we are minimizing $H(D, E, L)$, we can add a constant $C$ to the expression without changing the result:

$$\argmin_T H(D, E, L) = \argmin_T [H(D|L) + H(E|L) + H(L) + C]$$  \hspace{1cm} (A.14)

where $T$ is the camera calibration matrix. The luminance values in the optical image remain constant for the registration process, so we can let $C = H(L)$. This leads to

$$\argmin_T H(D, E, L) = \argmin_T [H(D|L) + H(E|L) + H(L) + H(L)]$$  \hspace{1cm} (A.15)

$$= \argmin_T [H(D, L) + H(E, L)].$$  \hspace{1cm} (A.16)
Appendix B

Pseudocode

The following code is a condensed form of the non-foliage multi-view algorithm that is described in 5.2.2. The algorithm uses LIDAR elevation data, a Delaunay mesh, knowledge of foliage points, and an existing registered image. We provide this code because it corresponds to our most complicated registration procedure; the other algorithms can be derived from this one fairly easily. The code is intended to show the basic calculations involved in the algorithm as well as the method for utilizing OpenGL. The code resembles MATLAB code, although many functions do not actually exist, such as all of the OpenGL functions.

function automaticRegistration

[objVal] = function calcObjectiveValue(cameraStruct) % defined below
[je] = function measureJointEntropy(imLadar, imPhoto) % defined below

%%%% Read files %%%
ladarData = open ladarFile ;
    % read LIDAR file, ladarData is an array of dimension
    % nPoints x 4 where each row is a [x,y,z,p]
    % value.  p is the probability of detection
    % for each point; it is an int of the range
    % [0,255].  ladarData is in single format.
mesh = open meshFile;
% Read mesh file. This file is obtained from performing
% a Delaunay triangulation with 10% of the LIDAR
% points. It contains sequential indices for the triangles,
% where each row is a [x,y,z] value, and there are
% three rows per triangle.

isFolige = open foliageFile
% isFoliage has dimension [nPoints, 1] and has zero
% values for non-foliage points and has a value
% of 1 for foliage points

imPhotoColor = open photoFile;
% read RGB image file for image to be registered.
% imPhotoColor is an array of
% dimension nPixels x 3 where each row is a
% [R,G,B] value

imRegColor = open regFile;
% open existing registered image file; same type as
% imPhotoColor

% Also load details such as initial registration,
% alignment for existing registered image

load regImageCameraStruct; % this is used implicitly
load initialCameraStruct;
% coarse starting point for image to be registered

% Preliminary calculations

global imPhoto = rgb2gray(imPhotoColor);
% calculate luminance; maintain array size
% this is a global variable; all
% functions can access it
imReg = rgb2gray(imRegColor); % same for existing registered image

nPoints = size(ladarData, 1); % determine number of LIDAR points

elev = ladarData(:,3); % scale elevation values to range [0,255]
elev = elev - min(elev);
elev = elev ./ max(elev);
elev = im2uint8(elev);

ladarIntensity = uint8(zeros(nPoints, 3));
    % Color encoding of point cloud for
    % rendering. Each row is an [R,G,B] uint8

ladarIntensity = [elev, elev, elev];
    % Grayscale intensities for elevation

foliageMarker = uint8([0, 255, 0]); % Color for foliage points (green)

for (n = 1:nPoints)
    if (isFoliage(n))
        ladarIntensity(n,:) = foliageMarker;
        % make foliage pixels green
    end
end

%%%% OpenGL inits %%%% 
openGL.setBackgroundColor('Blue'); % indication for background pixels
openGL.setPointSize(1.0); % pixel size for point rendering
openGL.setPointVertextBufferObject(ladarData(:,1:3));
    % upload onto graphics card
openGL.setlntensityVertextBufferObject(ladarIntensity);
    % upload onto graphics card
openGL.setMeshVertextBufferObject(mesh); % upload onto graphics card

%%%% Perform registration %%%% 
finalCameraStruct

117
= downhillSimplex(@calcObjectiveValue, intitalCameraStruct) 
% intitalCameraStruct and finalCameraStruct are each 
% 7 x 1 vectors with variables Cx, Cy, Cz, alpha, beta, 
% gamma, and fov. Theses are the parameters that are 
% estimated via optimization

save finalCameraStruct % Save final position

% Clear variables, etc....
end

[objVal] = function calcObjectiveValue(cameraStruct)
% Calculates objective value for registration. This function 
% is passed to the downhill simplex optimization routine.

  openGL.setCameraPosition(cameraStruct);

  imLadarPoints = openGL.renderPointCloud;
    % render point cloud encoded with ladarIntensity

  jePoints = measureJointEntropy(imLadarPoints);
    % obtain first joint entropy value

  imLadarMesh = openGL.renderMesh;
    % render mesh with registered image draped on
  jeMesh = measureJointEntropy(imLadarMesh);

  objValue = jePoints + jeMesh;
end

[je] = function measureJointEntropy(imLadar, imPhoto)
% Measures the entropy between imLadar and imPhoto
% imLadar and imPhoto are of the dimension nPixels x 3
% where each row is a [R,G,B] value. All values in
% the array are ints in the range [0,255]
% imPhoto is a global variable

pmf = zeros(256, 256) ; % initialize
nPixels = size(imLadar, 1) ; % determine number of pixels

for n = 1:nPixels

    ladarPixelRed = imLadar(n, 1) ;
    ladarPixelGreen = imLadar(n, 2) ;
    ladarPixelBlue = imLadar(n, 3) ;

    photoPixelRed = imPhoto(n, 1) ;
    % R=G=B for all photo pixels
    photoPixelGreen = imPhoto(n, 2) ;
    photoPixelBlue = imPhoto(n, 3) ;

    if(ladarPixelRed == ladarPixelBlue &&
        ladarPixelRed == ladarPixelGreen) {
        % make sure pixel is not foliage or background

            l = ladarPixelRed + 1 ;
            % R=G-B, so use any color to get intensity
            p = photoPixelRed + 1 :
            % R=G-B, so use any color to get intensity
            pmf(l,p) = pmf(l,p) + 1.0 ;
            % Indexing starts at 1

    end

end

pmf = pmf./sum(pmf(:)) ; % noramlize so sum(pmf) = 1
pmf = pmf(:) ; % turn into vector
pmf = pmf(pmf>0); % only use nonzero values for entropy calculation
je = sum(-pmf.*log2(pmf));
end
Bibliography


122


