DATA REGISTRATION WITHOUT EXPLICIT CORRESPONDENCE FOR ADJUSTMENT OF CAMERA ORIENTATION PARAMETER ESTIMATION

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Gabor Barsai, B.Sc., M.Sc.
Graduate Program in Geodetic Science

* * * *

The Ohio State University

2011

Dissertation Committee:

Prof. Alper Yilmaz, Adviser
Prof. Carolyn Merry
Prof. Ron Li
Prof. Alan Saalfeld
Abstract

Creating accurate, current digital maps and 3-D scenes is a high priority in today’s fast changing environment. The nation’s maps are in a constant state of revision, with many alterations or new additions each day. Digital maps have become quite common. Google maps, Mapquest and others are examples. These also have 3-D viewing capability. Many details are now included, such as the height of low bridges, in the attribute data for the objects displayed on digital maps and scenes. To expedite the updating of these datasets, they should be created autonomously, without human intervention, from data streams. Though systems exist that attain fast, or even real-time performance mapping and reconstruction, they are typically restricted to creating sketches from the data stream, and not accurate maps or scenes. The ever increasing amount of image data available from private companies, governments and the internet, suggest the development of an automated system is of utmost importance. The proposed framework can create 3-D views autonomously; which extends the functionality of digital mapping. The first step to creating 3-D views is to reconstruct the scene of the area to be mapped. To reconstruct a scene from heterogenous sources, the data has to be registered: either to each other or, preferably, to a general, absolute coordinate system. Registering an image is based on the reconstruction of the geometric relationship of the image to the coordinate system at the time of
imaging. Registration is the process of determining the geometric transformation parameters of a dataset in one coordinate system, the source, with respect to the other coordinate system, the target. The advantages of fusing these datasets by registration manifests itself by the data contained in the complementary information that different modality datasets have. The complementary characteristics of these systems can be fully utilized only after successful registration of the photogrammetric and alternative data relative to a common reference frame. This research provides a novel approach to finding registration parameters, without the explicit use of conjugate points, but using conjugate features. These features are open or closed free-form linear features, there is no need for a parametric or any other type of representation of these features. The proposed method will use different modality datasets of the same area: lidar data, image data and GIS data. There are two datasets: one from the Ohio State University and the other from San Bernardino, California.

The reconstruction of scenes from imagery and range data, using laser and radar data, has been an active research area in the fields of photogrammetry and computer vision. Automatic, or just less human intervention, would have a great impact on alleviating the “bottle-neck” that describes the current state of creating knowledge from data. Pixels or laser points, the output of the sensor, represent a discretization of the real world. By themselves, these data points do not contain representative information. The values that are associated with them, intensity values and coordinates, do not define an object, and thus accurate maps are not possible just from data. Data is not an end product, nor does it directly provide answers to applications, although implicitly, the information about the object in question is contained in the data. In some form, the data from the initial data acquisition by the sensor has to
be further processed to create useable information, and this information has to be combined with facts, procedures and heuristics that can be used to make inferences for reconstruction. To reconstruct a scene perfectly, whether it is an urban or rural scene, requires prior knowledge, heuristics. Buildings are, usually, smooth surfaces and many buildings are blocky with orthogonal, straight edges and sides; streets are smooth; vegetation is rough, with different shapes and sizes of trees, bushes. This research provides a path to fuse data from lidar, GIS and digital multispectral images and reconstructing the precise 3-D scene model, without human intervention, regardless of the type of data or features in the data. The data are initially registered to each other using GPS/INS initial positional values, then conjugate features are found in the datasets to refine the registration. The novelty of the research is that no conjugate points are necessary in the various datasets, and registration is performed without human intervention.

The proposed system uses the original lidar and GIS data and finds edges of buildings with the help of the digital images, utilizing the exterior orientation parameters to project the lidar points onto the edge extracted image/map. These edge points are then utilized to orient and locate the datasets, in a correct position with respect to each other.
Dedicated to my family...
ACKNOWLEDGMENTS

I am really honored to thank my mother and father for sacrificing their luxuries to give me proper education. Without their support, I would have never received a chance to perform higher studies; and my sister and niece, for always encouraging me.

It is a great pleasure to be a student at The Ohio State University, USA. I wish to thank all faculty and staff members who gave a chance to perform my studies here. I am really glad to convey my sincere thanks to Dr. Toni Schenk, for accepting my application. I thank him for explaining each and every aspect of photogrammetry at all times. He plays a vital role in the success of this dissertation.

I would like to extend my thanks to Dr. Alper Yilmaz for exposing me to the computer vision concepts and for his contagious enthusiasm. His courses helped me to realize the importance of integrating both computer vision and photogrammetry perspectives in solving mapping problems. I also thank him for his invaluable advices to improve my dissertation.

My sincere thanks to Dr. Ron Li, Dr. Carolyn Merry, Dr. Alan Saalfeld and Dr. Aravind Asthagiri for being in my dissertation committee. Their suggestions to improve the dissertation methodology as well as the report was very helpful.
I am grateful that Dr. John Bossler, Dr. Clyde Goad Dr. Ivan Mueller and Dean Ringle kept in touch with me for such a long time, and for Dr. Dorota Grejner-Brzezinska, for guiding my application through the department, and to Hunagi and T.U. of Budapest, for keeping in touch, and to the McCoy family in Westerville for all their kindness.

I also thank former and present Photogrammetric Computer Vision Lab members Ekta Amar, Sagar Deshpande, Yushin Ahn, Taehun Yoon, Wonhee Lee, Po-Lun (Ryan) Lai, Sudhagar Nagarajan, Sajid Mahmood, Yeon Yeu, Kyoungjin Park, Panu Srestasathierin, Youngjin Lee, Heewon Lee, Vinod Khare, Rhae-Sung Kim, Drew Kerns, Anuchit Sukcharoenpong, Adem Hayal, Sibel Uzun, Boris Skopljak, Jinwei Jiang, Mohammed Al-Shahri, Mustafa Ozendi and Daniya Zamalieva for discussing their research ideas as well as for giving insightful comments to improve my ideas, and to the members of the GIS and GPS labs,

and

Ella: “Life is not a puzzle to be solved, but a gift to be enjoyed”.
Gabor Barsai has worked in the fields of surveying and mapping as a GPS (Global Positioning System) and GIS (Geographic Information Systems) professional. His GPS experience includes geodetic control surveys, boundary location and topographic surveys where he has served as instrument operator and in positions of responsible charge, and as application developer to process the data; as a GIS professional, Mr. Barsai has served as a GIS consultant, GIS manager, and application developer with international experience in both GPS and GIS (including teaching and conference presentations and simultaneous translator for several organizations). Mr. Barsai is a graduate of the Ohio State University with a Master of Science in Mapping/GIS, and is a current Ph.D. candidate in the Civil Engineering Dept. studying Digital Mapping. His primary research interest is facilitating the way in which different data modalities interact with each other and fuse themselves as needed to create a more accurate estimate of the environment. Research work focuses on exploring methods and developing tools to allow for dynamic application of data fusion in multiple environments, with the goal of allowing applications to fully exploit the potential of
such different modalities and facilitating this task for future applications development. Applications include environment monitoring, surveillance, automatic target detection and tracking, remote sensing. Data fusion research has revolutionized the way science is being done today and has opened up a vast range of opportunities for solving complex problems that scientists could not even think about solving in the near past. Data fusion will support automated feature extraction, which has long been a goal for the imaging industry. To date no method has been perfected to automatically extract features such as building footprints, edge of roads and walkways, vegetation, or bodies of water. The process requires very high-resolution imagery, with analysis performed largely in a 2D environment requiring significant manual editing. Incorporating 3D data and information into the process greatly aids in the analysis and extraction efforts. Like the imaging industry, range data providers have faced similar issues. Spatial analysis alone is often inconclusive when attempting to determine whether a range point has hit a small bush, fire hydrant, boulder or a ground surface anomaly. Conventional surface classification filters can only go so far when removing above-ground phenomena, as natural and artificial objects may be spatially interpreted similarly and subjectively removed, inadvertently eliminating valid surface detail without differentiation.

PUBLICATIONS

Research Publications


FIELDS OF STUDY

Major Field: Geodetic Science

Area of emphasis:

Topic 1 Digital photogrammetry
Topic 2 Object recognition/Feature Extraction
Topic 3 Scene/Surface reconstruction
Topic 4 GIS
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>v</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>vi</td>
</tr>
<tr>
<td>Vita</td>
<td>viii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xiv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xvi</td>
</tr>
</tbody>
</table>

### Chapters:

1. Introduction ................................................. 1

2. Background .................................................... 11
   - 2.1 Terminology ............................................ 11
   - 2.2 Background ............................................. 12
   - 2.3 Points vs. Features .................................... 27
   - 2.4 Registration methods ............................... 31
     - 2.4.1 Registration of images ....................... 31
     - 2.4.2 Registration of laser scanning data ........ 32
     - 2.4.3 Registration of maps/GIS .................... 32
   - 2.5 Point based registration models ............... 33
     - 2.5.1 2-D similarity registration............... 33
     - 2.5.2 Affine transformation .................... 34
     - 2.5.3 3-D registration ............................ 35

xi
5. Feasibility studies ........................................ 123
   5.1 Data description ...................................... 124
      5.1.1 Camera orientation parameters from GIS data combined with lidar elevation data ........................................ 132

6. Implementation and Results ................................ 146
   6.1 Implementation ........................................ 146
      6.1.1 DT Convergence .................................... 148
      6.1.2 Camera orientation parameters from lidar only data ........ 150
      6.1.3 Extraction of building outline from lidar data .......... 152
      6.1.4 Harris corner detection ............................. 153
      6.1.5 Hough transform .................................... 154
      6.1.6 Relative orientation ................................. 156
   6.2 Results ................................................ 157
      6.2.1 Camera orientation from lidar only and from GIS and lidar 157
      6.2.2 Bundle adjustment of images ......................... 166
      6.2.3 Relative orientation ................................. 166
      6.2.4 Discussion ........................................... 170

7. Summary and Future Work ................................ 184

Appendices:

A. Draping imagery over lidar data results .................. 188

B. Exterior orientation chart ................................ 196

Bibliography .................................................. 201
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Matching method and entities</td>
<td>29</td>
</tr>
<tr>
<td>5.1</td>
<td>ALTM3100 parameters</td>
<td>126</td>
</tr>
<tr>
<td>6.1</td>
<td>Image 425 solutions</td>
<td>160</td>
</tr>
<tr>
<td>6.2</td>
<td>Image 426 solutions</td>
<td>161</td>
</tr>
<tr>
<td>6.3</td>
<td>Image 427 solutions</td>
<td>162</td>
</tr>
<tr>
<td>6.4</td>
<td>Mean distances computed in meters for X, Y and Z coordinates between lidar points and recovered 3D points for image 425 and 426 of the California study site. The recovered points are estimated by intersecting the rays from corresponding points between two images whose camera parameters are estimated using the proposed method.</td>
<td>163</td>
</tr>
<tr>
<td>6.5</td>
<td>Solutions for (a) the principal point offset and focal length all in millimeters; (b) perspective center coordinates in meters; and (c) camera orientation in degrees. The results are generated for the Ohio State University study site.</td>
<td>164</td>
</tr>
<tr>
<td>6.6</td>
<td>Mean distances computed in meters for X, Y and Z coordinates between lidar points and recovered 3D points for the OSU study site. The table includes results using initial parameters and adjusted parameters for manual and proposed approach.</td>
<td>165</td>
</tr>
</tbody>
</table>
6.7 Mean distances computed in meters for X, Y and Z coordinates between lidar points and recovered 3D points for the California study site, using proposed approach with GIS. The table includes results using adjusted parameters, using points matched to the exact lidar elevation. (a) Image 425 (b) Image 426 (c) Image 427.

6.8 Bundle Adjustment solutions

6.9 Estimated relative orientation parameters and their standard deviations for manual, SIFT based and proposed approaches for Model 1. The unit of reported values are in millimeters. Boldface indicates best results with least deviation.

6.10 Model 1 photo coordinate residuals for left and right images from (a) SIFT matching, (b) DT matching, (c) manual matching:

6.11 Estimated relative orientation parameters and their standard deviations for manual, SIFT based and proposed approaches for Model 2. The unit of reported values are in millimeters. Boldface indicates best results with least deviation.

6.12 Model 2 photo coordinate residuals for left and right images from (a) SIFT matching, (b) DT matching, (c) manual matching:

6.13 Estimated relative orientation parameters and their standard deviations for manual, SIFT based and proposed approaches for OSU Model. The unit of reported values are in millimeters. Boldface indicates best results with least deviation.

6.14 Mean square error, $\epsilon$, computed for x and y coordinates in millimeters in the image space.
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>EO parameters: $X_c, Y_c, Z_c, \omega, \phi, \kappa$ with image point $a$ [25]</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>IO parameters: $x_0, y_0, f$. $PP$ stands for principal point, $f$ is shown as negative, because in a right-handed system the perspective center has a negative $z$ coordinate value [25]</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>The inverse collinearity equation needs a reference elevation.</td>
<td>38</td>
</tr>
<tr>
<td>2.4</td>
<td>Active contour example. Black nodes are manually selected nodes on road, white nodes are detected edge points on road via active contour method. The manually selected black nodes were moved by the function to the road edge.</td>
<td>51</td>
</tr>
<tr>
<td>3.1</td>
<td>(a) Distance transform, $D$, computed from a closed edge segment superimposed as white line. Larger distances have lighter shade. (b) Euclidean distances computed from a central pixel that has a binary value of 1 in a 5x5 neighborhood. Image (a) represents the actual image, (b) represents the values of the distance transform. The left side of (b) represents an edge, where the edge pixels are represented as a binary value of 1, non-edge pixels are shown as 0. The right side of (b) shows the result of the distance transform, where the numbers represent the closest distance to the edge points on the left side.</td>
<td>62</td>
</tr>
<tr>
<td>4.1</td>
<td>Process flow diagram of the proposed method for exterior orientation</td>
<td>78</td>
</tr>
<tr>
<td>4.2</td>
<td>GPS/IMU/camera setup [77]</td>
<td>83</td>
</tr>
</tbody>
</table>
4.3 Effect of radial distortion on the image geometry. Dashed line represents the rectangular object as it would appear in absence of radial distortion. The solid line shows the object shape in the presence of (a) barrel and (b) pincushion distortion. [160] 

4.4 The probability surface of a building from image 425 Hough edge map; the Hough transformed image is shown on fig. (6.2). Each edge is fitted or superimposed with a normal distribution, to represent the uncertainty in the edges. Red color represents probabilities closer to edges, blue represents probabilities less likely to be edges; the vertical coordinate represents the probability, with the maximum of 1. 

5.1 Site of data collection in San Bernardino, CA [58] 

5.2 (a) Image 425 and (b) Canny edge detection (c) Edge detection after segmentation by EDISON/mean shift 

5.3 (a) Image 426 and (b) Canny edge detection (c) Edge detection after segmentation by EDISON/mean shift 

5.4 (a) Image 427 and (b) Canny edge detection (c) Edge detection after segmentation by EDISON/mean shift 

5.5 Raw lidar of California area, showing the Welcome Center. Correct \((x, y, z)\) positions 

5.6 (a) California welcome center (CWC on street map) [58] (b) Location with respect to Google Maps 

5.7 (a) Image 40-01 used in the experiments for the OSU study site. The “Horse Shoe” shaped Ohio stadium is on the middle left of the image. (b) Image 39-014 used in the experiments for the OSU study site. The “Horse Shoe” shaped Ohio stadium is on the top left of the image. 

5.8 (a) OSU GIS data (cropped for better visualization) that corresponds to part of the image in part (a). The GIS data contains the buildings, street centerlines and curbs; note that the stadium on the middle left. (b) Local lidar elevation image for the same study site with stadium in middle left. Only local data is shown for better visualization.
5.9 (a) Elevation thresholded lidar data for San Bernardino (b) GIS of San Bernardino local area with buildings, streets centerlines, curbs, parcels. 143

5.10 Ideal case for GIS and lidar data registration with no noise. GIS and lidar data from different sources register to each other quite well. Usually, this is not the case, since trees and lamps interfere and make lidar returns difficult to distinguish. (a) Near the Ohio State campus, the rectangular features represent the lidar data from the Ohio Geographically Referenced Information Program (OGRIP). The building outlines are from the Franklin County Auditor’s Office GIS. (b) Welcome Center Building near San Bernardino freeway, visible on images 425-427. 144

5.11 (a) Cropped image from the California study site image 425. (b) Edges detected using the Canny edge detector. Notice that some edges are not complete. (c) Closed edges detected using mean-shift segmentation. 145

6.1 High elevation variability filtered lidar. (a) White areas show constant slopes, black areas show variable slopes for an area. The black areas match vegetation and cars in the parking lot: non buildings. (b) Lidar data. Different elevations are in different colors. (c) Overlaid on lidar data. 174

6.2 (a) Filtering of linked edges by applying a Hough transform and 80 pixel length threshold (see text for details). (b) lidar edges projected to image space using initial GPS/IMU values for image 425. 175

6.3 (a) Input image for Hough line transform after edge detection. (b) Only straight lines are of interest. 176

6.4 Process flow diagram of the proposed method for RO. 176

6.5 Subset of conjugate keypoints detected and match using the scale invariant feature transform (SIFT). The ends of the green lines represent conjugate points. (a) California images 425, left, and 426, right (b) OSU images 40-01, left, 39-014, right. 177

6.6 (a) Gaussian noise added to GIS data (b) original building. This is the building on the right, in the middle of fig.(5.9) 178
6.7 Comparison of lidar points projected onto image 425 before and after adjustment of lidar data. Green pixels represent the lidar points, red outline used to enhance visibility. Local area is zoomed in to show the improvement. Projection using (a) initial camera parameters, (b) after coarse step, and (c) after refined adjustment step. 179

6.8 Coarse to fine parameter estimation step by minimizing (6.11) applied to GIS ground data. Local area is zoomed in to show the improvement. Projection using (a) initial camera parameters, (b) after coarse step, and (c) after refined adjustment step. 180

6.9 Draping comparison of various methods for Image 425. 181

6.10 Bundle adjustment of all images overlaid on GIS. 182

6.11 California data relative orientation. Projection of the edges in target image and the reference image using (a) initial parameters, (b) final precise parameters. Note the increased overlap between projected edges and the reference image edges. 183

7.1 Image 425 draped over level set representation of lidar points. 187

A.1 Draping comparison of various methods for Image 425. 189

A.2 Draping comparison of various methods for Image 426. 190

A.3 Draping comparison of various methods for Image 426. 191

A.4 Draping comparison of various methods for Image 427. 192

A.5 Draping comparison of various methods for Image 427. 193

A.6 Draping comparison of various methods for bundle adjustment. 194

A.7 Draping comparison of various methods for bundle adjustment. 195

B.1 Adjustment results 425. 198

B.2 Adjustment results 426. 199

B.3 Adjustment results 427. 200
Photogrammetry is the first remote sensing technology ever developed in which geometric properties about objects are determined from photographic images. Photogrammetry (literally, “measurements from photographs”, while photograph literally means “writing with light” from the greek term photo=light and graph=write) is as old as modern photography and can be dated to the mid-nineteenth century. Remote sensing, as the name implies, deals with the measurement or acquisition of information of some property of an object or phenomenon by a recording device that is not in physical or intimate contact with the object or phenomenon under study [47]. While aerial and terrestrial photographs are still used in many applications, the field of photogrammetry is shifting from traditional film-based technologies to digital image based ones. This new trend, called digital photogrammetry or softcopy photogrammetry, is a field that is closely related to the vision technologies with the common ultimate goal of 3-D perception [143]. Since it is an emerging field, an exact definition for softcopy photogrammetry or a softcopy photogrammetric workstation, basically a computer designated for softcopy photogrammetry, is still evolving. However, it is widely accepted that a softcopy photogrammetric workstation includes image processing, three-dimensional coordinate calculation and other functionalities, and operates
interactively and automatically toward the completion of photogrammetric tasks [39]
[143].

One of the many tasks of photogrammetry is the reconstruction of 3-D scenes
and mapping from aerial photographs, successfully accomplished by skilled human
operators working in an interactive environment with analytical plotters or softcopy
workstations. The goal of scene reconstruction is to determine a scene $S$ that approx-
imates an unknown scene $S_u$, using data from some sampling method and information
about the sampling process, which includes noise, resolution and the data collection
method. The reconstructed scene provides a surface of the sampled area. It has
long been recognized that surfaces and their properties and characteristics play an
important role in image understanding and object recognition [148]. Many other
perception tasks, such as navigation and urban reconstruction make extensive use of
surface properties. Surfaces are intermediate representations in the long processing
chain from data to objects. Automating this, and other processes like registration, is
the ultimate goal of digital photogrammetry, which is to autonomously create maps
from data. Surface reconstruction from unorganized datasets is very challenging in
three, and/or higher, dimensions. The problem is ill-posed, i.e, there may be no
unique solution due to occlusion, missing or incomplete data. Furthermore, the or-
dering or connectivity of datasets and the topology of the real surface can be very
complicated in three, and/or higher, dimensions. With the ever increasing amount
of data and the need to deliver results quickly and economically, research during the
last decade focused on automating photogrammetric processes. For accurate surface
reconstruction, data fusion may create better results. While aerial images provide ac-
curate positional information, their information content on surface material properties
is limited. Combining an accurate surface with correct spectral and radiometric properties should render a realistic scene. Data fusion necessitates that the various data sources be registered to each other. The preliminary step to reconstruction or data fusion is registration of the various datasets to each other: aligning the features and objects to each other, for example, so that a house in one dataset is aligned with the same house in another dataset. This way, complementary information in each dataset may be used with the other dataset, to fully exploit the information. Registration is the process of determining the geometric transformation parameters of a dataset in one coordinate system, the source, with respect to the other coordinate system, the target. The dataset can be in the form of pixels, points, lines or surfaces. Registering data to a known coordinate system, like State Plane, is known as georeferencing. To register datasets to each other, either conjugate features or objects are required, then matched to each other to find the transformation parameters; or the orientation of the sensor that captured the data must be known in some coordinate system. Using either method, the reconstruction of the geometric relationship of the datasets to each other or a coordinate system is possible. If the reconstruction of the geometric relationship of the datasets is just to each other, it is a relative transformation; if it is to a known coordinate system, it is absolute. If the absolute transformation is known, then multiple datasets can be registered to each other, as the parameters of the transformation are already given. Of course, even a relative transformation can give absolute coordinates, if one of the datasets is in a known coordinate system.

The problem of interest can be stated as follows: given partial information of an unknown scene from different data modalities, construct a representation of the scene by registering the datasets to each other. A general registration formulation can be
stated as: given two datasets, an input $\mathcal{D}$ and a target $\mathcal{S}$, and a dissimilarity measure, find the best transformation that associates to any point of $\mathcal{D}$ a corresponding point at $\mathcal{S}$ and minimizes the dissimilarity measure between the transformed $\hat{\mathcal{D}}$ and the target $\mathcal{S}$. This dissimilarity can be defined either as local, by extracting similar shapes, or global for the entire dataset [124]. Global models are assumed to be valid for the entire region/image to be registered. On the other hand, local deformations refer to pixel- or point-wise registration components. Registration problems occur in diverse scientific and engineering application domains, including [72]:

- Registration of stereo pairs: A stereo pair is a pair of photos of the same area from slightly different angles. If the photos are oriented as at the time they were taken, then a virtual 3-D model can be established. The idea is similar to how the human vision works. Because the eyes of humans and other highly evolved animals are in different positions on the head, they present different views simultaneously. This is the basis of stereopsis, the process by which the brain exploits the parallax due to the different views from the eye to the target object to gain depth perception and estimate distances to objects. To orient the stereo pairs relative to each other, conjugate points or lines are necessary. Traditionally, conjugate points/lines are found by human operators in photogrammetry.

- Registration of range data: The data produced by laser range scanning systems is typically an irregular grid of distances from the sensor to the object being scanned. If the sensor and object are fixed, only objects that are “point viewable” can be fully digitized. More sophisticated systems, such as those produced by Cyberware Laboratory, Inc. ¹, are capable of digitizing cylindrical objects.

¹http://www.cyberware.com
by rotating either the sensor or the object. However, the scanning of topologically more complex objects, including those as simple as a coffee cup with a handle, cannot be accomplished by either of these methods. To adequately scan these objects, multiple viewpoints must be used. Merging the data generated from multiple viewpoints to reconstruct a polyhedral surface representation is a non-trivial task [147].

- **Registration of contours:** Contours are the traditional way to represent elevation or features, like buildings or lakes, on paper maps for a 2-D surface. A United States Geological Survey (USGS) topographic map contains contours, in fact it is one of its most distinguishing features. In many medical studies it is common to slice biological specimens into thin layers with a microtome. The outlines of the structures of interest are then digitized to create a stack of contours. The problem is to reconstruct the three-dimensional structures from the 2-D contours.

Creating accurate maps is not a new idea: “Also you must note, there are diverse fashions of landes, and therefore diversly to be measured. And some manner of lande lieth in suche sondrie formes, that it must needes be measured, not in and whole, but in divers parcelles, every parte by itself” [96]. The book by Leigh is one of the many books from the late Renaissance period with a new awareness of the need for greater accuracy and precision in the description and measurement of land. Increasing land values and trade required accurate mapping [165]. The revolution in surveying which took place from 1550 to 1650 was associated with three contemporary developments: an increasing interest in navigation with its accompanying application of astronomical theory; progress in the military sciences coupled with the application of geometry
to the problems of sighting; and the agrarian changes of the period together with the improved methods of farming [36]. One could argue that the revolution could be pushed back even further, to 1450, coinciding with the “Age of discovery”, and the necessary mapping of the Americas and Africa, and later Asia and Australia. However, this is beyond the scope of the current subject. Nearly 450 years after Leigh’s book, accurate and precise mapping is still a necessary requirement for public and private entities, agriculture and industry, engineering, although significant changes have occurred in the surveying field, from remote sensing, aerial photography to laser scanning, to computer mapping/geographic information systems (GIS). The current mapping revolution is due to a combination of factors, like in Leigh’s time, which include advancements in navigation (Global Positioning System (GPS), Inertial Measurement Unit (IMU)), accurate city/land databases for GIS.

Developing registration methods without the need for human interaction has been an ongoing research activity [112] [145] [88] [161] [159]. Registration algorithms addressing these problems have typically been crafted on a case-by-case basis. For instance, algorithms solving the registration of contours problem make heavy use of the fact that data are organized into contours, i.e., closed polygons, and that the contours lie in parallel planes.

The selection of the registration model is also related with the transformation domain. Using a relative transformation model, local registration domains are suitable for objects that undergo local deformations, for a non-rigid transformation. Correspondences across small neighborhoods are introduced to ensure the regularity of the registration. These methods exhibit strong sensitivity to noise. One can claim that their ability to deal with non-rigid motion makes them quite attractive.
Opposite to the local registration methods, global transformations are valid for the entire data. Correspondences between the target $D$ and the source $S$ are obtained by applying the same transformation to the entire data. Robustness is the key characteristic of these methods. The estimation of the registration parameters can be done reliably due to the high number of measurements from conjugate pixels/points available for solving the inference problem. On the other hand, such methods perform poorly if the assumptions related with the registration problem are not satisfied. This is defined as non-rigid motion. Neither the selection of the transformation, nor the domain are sufficient enough to determine the registration. These two components, local and global, have to be integrated in an optimization framework by means of selecting a dissimilarity measure that involves the source and the target shape, the motion model and the registration domain [124].

This research provides a novel approach to finding registration parameters. The proposed approach is to pose a unifying general solution that does not assume any structure on the data points for registration. Photogrammetric registration methods are point based [146]. This works well in an office environment with trained photogrammetrists finding conjugate points, but is difficult to automate with different modality datasets. The purpose of this research is therefore to address this registration problem by introducing an image registration approach that does not rely on individual points, but linear features, such as edges and curves, and can also deal with local deformations, such as occlusion or missing data, while taking into account the entire dataset.
After correct registration, the fused datasets may be used for applications, which include environmental monitoring, surveillance, automatic target detection and tracking and remote sensing. Data fusion research has revolutionized the way science is being done today and has opened up a vast range of opportunities for solving complex problems that scientists could not even think about solving in the past. Future applications will support automated feature extraction, which has long been a goal for the imaging industry. To date no method has been perfected to automatically extract features, such as building footprints, edge of roads and walkways, vegetation, or bodies of water. The process requires very high-resolution imagery, with analysis performed largely in a 2-D environment, requiring significant manual editing. Incorporating 3-D data and information into the process greatly aids in the analysis and extraction efforts, however, like the imaging industry, range data providers have faced similar issues: spatial analysis alone is often inconclusive when attempting to determine whether a range point has hit a small bush, fire hydrant, boulder or a ground surface anomaly, since conventional surface classification filters, when removing above-ground phenomena may be subjectively removed, inadvertently eliminating valid surface detail without differentiation.

While maps are now used everywhere, there is still a general misunderstanding about them: “The greatest enemy of progress is ignorance and I think I am safe in calling maps the most used and least understood documents of modern civilization. The tendency may be to believe everything that appears on a map without understanding the limitations of maps and knowing how to use them” [23]. While not the theme of this research, map understanding is perhaps the next step for cartography, after map creation.
The following briefly describes the organization of the dissertation:

Chapter 2 proved a description of the advantages of registering features over points. The terminologies used in this research and other literature are also elaborated upon, to introduce the idea of an automatic registration method, and to compare the proposed method with other methods. The terminology section provides a discussion of the corresponding mathematical models. It also contains an overview of available techniques. Finally, different registration methods are compared.

Chapter 3 elaborates on the proposed method of automatic registration and the justification behind the approach. It briefly discusses an overview of mathematical models. Functional models for the method are discussed in the respective sections. It also broadly discusses the assumptions made in the implementation of the algorithms. It also adds the significance of this approach.

Chapter 4 contains the implementation of the proposed method. It further discusses the results of experiments with synthetic and real data and proves the capability of the algorithm. Results are analyzed and the feasibility of the algorithm is discussed.

In Chapter 5 the proposed method of automatic registration is adapted to the well known surface reconstruction problem in photogrammetry and the data are described. Feasibility of the method is elaborated.

Chapter 6 provides a discussion for adjusting the data. In addition to that, an implementation of the algorithm is discussed with several experiments.

Chapter 7 summarizes the advantages, disadvantages and limitations of the proposed method of automatic registration. Recommendations are made to overcome
the limitations of the proposed method. Finally, how the introduced idea could be extended to other problems in photogrammetry is discussed along with future work.
Chapter 2

BACKGROUND

2.1 Terminology

This section explains the terminologies often used in this dissertation with some examples. In this work, data is a general term, it can be a linear feature or point based. Feature refers to area or linear primitives, although a linear feature is composed of points, and an object refers to a collection of linear and point features with similar characteristics. A linear feature has only length, and no width, and it can take on any shape or curve, not necessarily just a straight line. A free-form linear feature has no specific equation or derivation; in contrast to this are specific shapes, like conics, that have a specific equation and are derived from the intersection of a plane with a cone, hence the name “conic”. An areal feature can be any closed linear feature, it has a width and length. A surface feature is the outer boundary of some 3-D object, it can also be composed of several areal features. An image refers specifically to a digital format picture, while a photo is a hardcopy, traditional emulsion based photographic picture. Edges determined from edge detection on images are also considered linear features, independent of the edge detection method. An ideal edge is a discontinuation
of the spatial gray value function of the image domain [185]. This is true for the lidar
domain also, but using elevation values instead of pixel values.

Registration of aerial images is an important task in photogrammetry, as it is
a pre-requisite for surface reconstruction and orthophoto generation, as well as for
performing higher level inference tasks such as object recognition [146]. The scope of
photogrammetry is extended these days to reconstruction and recognition where linear
and surface features play a significant role; a mathematical model that directly uses
linear and surface features in registration will help in later stages of photogrammetric
processes such as reconstruction and recognition [143].

Reconstruction and recognition are dependent on registering the datasets correctly
to each other.

2.2 Background

Analytical photogrammetry is the application of mathematical principals to mea-
surements in photographic image planes in order to model and reconstruct both the
original spatial orientation of the image and the relative and absolute locations of
imaged object points. While the principals of spatial geometry basic to analytical
photogrammetry were seriously investigated as early as the 15th century by such
polymaths as Leonardo da Vinci and Albrecht Dürer, their application to the mea-
surement of photographs largely began with the work of Sebastian Finsterwalder at
the beginning of the twentieth century [41]. The development of analog measuring
devices that recreated the geometry of aerial photographs for the purposes of topo-
graphic mapping dominated the practical application of photogrammetry for the first
half of the twentieth century. The solution of the same problems analytically requires
an amount of computation that precluded practical application before the advent of computer processing. Otto von Grüber, who derived the projective equations fundamental to modern analytical photogrammetry, is famously quoted as saying that their application was “a waste of time and is of minor importance” [41].

Today large datasets are available from a number of sensors, gathering data in a variety of geometric, radiometric, temporal and thematic resolutions. In order to relate this data from heterogenous sources to each other they have to be registered - either to each other or, preferably, to a general, absolute coordinate system [153]. Registering an image is based on the reconstruction of the geometric relationship of the image to the coordinate system at the time of imaging in the coordinate system. Registration is the process of determining the geometric transformation parameters of a dataset in one coordinate system, the source, with respect to the other coordinate system, the target. The coordinate systems can be 2-D or 3-D and the transformation between the datasets can be 2-D or 3-D. The dataset can be in the form of points, lines or surfaces. Registering data to a known coordinate system, like State Plane, is known as georeferencing. Georeferencing, or even just registering photogrammetric products to other datasets, allows for the fusion of diverse datasets. Registration methods can be classified based on three aspects: representation of dataset, correspondence and a mathematical model [119]. The advantages of fusing these datasets by registration manifests itself by the data contained in the complementary information that images or photos do not have. Lidar (light detection and ranging) or radar (radio detection and ranging) data contain elevation data, which is lost during imaging, but lidar or radar data is inferior in visual information. GIS (geographic information system) data
is already processed and the points, lines and polygons all contain attribute data describing the various features, which is useful in object extraction, but is low in visual detail. While lidar data is not a product of any photogrammetric application, it is a very useful product when combined/fused with image data, and to a lesser degree this is true with radar data, but radar is, generally, less accurate [76]. Thus, the combination of range data and aerial data compensate each other with their disadvantages.

GIS data may be derived from photogrammetric data, and is also accompanied by data from other sources: surveys, notes, timetables, schedules. The complementary characteristics of these systems can be fully utilized only after successful registration of the photogrammetric and alternative data relative to a common reference frame.

Georeferencing of datasets may be achieved by finding conjugate features in the source and target coordinate systems, which leads to the transformation parameters involved. In photogrammetry, this involves the reconstruction of the geometric relationship of the source, the camera coordinate system, to the target coordinate system, a known coordinate system, at the time of data collection based on the orientation of the sensor: where the sensor is, which way it is oriented and details of the sensor itself; also known as the camera orientation parameters. Two types of camera orientation parameters need to be recovered, or known a priori, which is shown on Fig.(2.1) and Fig.(2.2):

1. **extrinsic camera orientation parameters**: The parameters that define the location and rotation of the camera reference frame with respect to a known world reference frame, also known as exterior orientation (EO). A single image or photo has six exterior orientation parameters (EOPs): three orientation angles, \( \omega, \phi, \kappa \), for orienting the camera axis at the instant the image was taken and the
three coordinates of the perspective center, $X_c, Y_c, Z_c$. The successive rotations of the sequence $\omega, \phi, \kappa$ or $\phi, \omega, \kappa$ are used. A transformation from one rotation system to the other can be made – the angular values are different, but the numerical values of the corresponding rotation matrix are identical [78]. These parameters are also known as “pose” in computer vision.

2. **intrinsic camera orientation parameters**: The parameters necessary to link the image coordinates of an image point with the corresponding coordinates in the camera reference frame, also known as interior orientation (IO). IO is achieved by a camera calibration process. IO certificates provided by camera manufacturers include the principal point offsets (PPO), which is the offset of the projection of the perspective center on to the image plane from the image center, the calibrated focal length and the radial symmetric lens distortion [78] and, sometimes, tangential lens distortion. The influence of the lens system is also investigated as part of the interior orientation process. sometimes broken into lateral and longitudinal chromatic aberration), spherical aberration, coma, curvature of field. An aberration is the failure of an optical system to bring all light rays received from a point object to a single image point or to a prescribed geometric position [104]. It is caused by the faulty grinding of the lens. Unlike distortion, aberrations do not affect the geometry of the image but instead affect image quality.

Interior orientation models the interior construction of a camera. It models a bundle of rays passing from the image points on the image plane through the corresponding perspective center. Exterior orientation models the camera location, actually the perspective center, and orientation of the camera axis at the time of imaging.
The line from the perspective center perpendicular to the image plane is called the principal axis or camera axis of the camera, and the point where the axis meets the image plane is called the principal point \[68\]. EO thus models the bundle of rays coming from the terrain to the perspective center. EOPs are usually defined in some agreed, absolute coordinate system, and are used to provide information for single images. To acquire accurate and precise measurements from the image, both IO and EO have to be known at the instant of exposure: EO models the camera perspective center with respect to the ground coordinate system, IO models the camera perspective center with respect to the image plane. Both IO and EO together will be referred to as camera orientation parameters. Knowing the camera orientation parameters is a prerequisite for any task involving the measurement of three-dimensional coordinates, such as the generation of a digital elevation model (DEM), the generation of

Figure 2.1: EO parameters: \(X_c, Y_c, Z_c, \omega, \phi, \kappa\) with image point \(a\) \[25\]
orthophotos, registration of datasets and the acquisition of data for geographic information systems, thus, recovering or knowing the camera orientation parameters is one of the most important problems in photogrammetry [69].

In order for the EOPs to be accurate and precise, the interior orientation parameters (IOPs) of the imaging camera must also be known, as there is a direct relation between these parameters and the EOPs. IOP values are given by the camera calibration report. The “Report of Calibration” provides quantitative camera system characteristics that are used to produce image and map products with a high degree of geometric accuracy and spatial resolution. This process provides for the following [10]:

Figure 2.2: IO parameters: \(x_0, y_0, f\). \(PP\) stands for principal point, \(f\) is shown as negative, because in a right-handed system the perspective center has a negative \(z\) coordinate value [25]
• Comparisons of the quality, condition, and any change of the camera system, from calibration to calibration.

• Information about camera characteristics for compliance with contract specifications.

• Camera system characteristics for correcting film imagery used in making accurate products.

• A standardized, repeatable test environment for repeated calibration.

IO is accomplished by several methods. In the Western Hemisphere, exposing the image plane through the camera body from multi-collimators is used for calibration of standard mapping cameras. In Europe, an optical approach termed the goniometer method is used. In either case, the procedure is conducted within a laboratory under closely controlled environmental conditions. Worldwide, other methods of camera calibration are used at various facilities and under various circumstances of application. Examples are the methods of plumb lines, convergent orientation, and three-dimensional control field. With these methods, some form of functional constraint is usually employed. Depending on application, these methods may be considered either component or system approaches to calibration [10].

For a single image, to determine EO parameters, there are two kinds of methods available. The traditional method of determining EOPs, using a set of Ground Control Points (GCPs) matched to the corresponding image points is called indirect orientation via spatial resection, or just indirect orientation. The GCPs, as the name suggests, are points on the ground/terrain whose coordinates are known in some coordinate system, and are used to locate and orient the images to the terrain. GCPs have
to be visible on the images in order to locate them. GCPs can be natural features or manually marked points on the terrain. In 1984, the Federal Geodetic Control Committee (FGCC) set up standards for the quality of horizontal and vertical control: first order, second order or third order [22]. First order points have the greatest accuracy. Of course, the required accuracy for a certain project is dictated by the user; not all work requires first order accuracy. The method of determining EOPs using onboard Global Positioning Systems (GPS) and Inertial Measurement Unit (IMU) observations is called direct orientation. Direct orientation is a relatively new concept, where the orientation parameters are determined in flight. Direct orientation, using GPS/IMU data may not always be available, due to cost or signal limitations. Even if direct orientation is available, the accuracy and precision of the result is questionable, due to camera model errors and GPS/IMU errors. A comparison of lidar and photogrammetric errors is presented in [59], [93] and [143]. These GPS/IMU errors can only be corrected by using GCPs, although in a more limited number than in indirect orientation. The direct orientation method determines EOPs of the camera using GPS and IMU measurements. The indirect registration method uses the collinearity equations to determine the EOPs of the camera from a set of GCPs. The collinearity equation and its usage in determining orientation are discussed in section 2.5.4.

The mathematical models of indirect orientation method are the collinearity condition, coangularity condition or coplanarity condition [60]. The collinearity model is based on the principle that perspective center, image point and corresponding ground point are collinear: they fall on the same 3-D line. This is the most used model in photogrammetry [66], defined by the collinearity equations. In the coangularity model,
the angle between two given points seen from the perspective center in the object space is equal to the angle between the images of these points, defined from the perspective center in the photograph. The coplanarity condition is based on conjugate points in images, since in most photogrammetric problems, object points are recorded on two or more photographs. For two photos, the two conjugate rays defined on each object point must be coplanar. The corresponding mathematical coplanarity condition implies that the two camera centers, the two image points, and the object point are in the same plane, the epipolar plane. The coordinates of the object point do not appear in the equation, so no approximations for the coordinates are needed.

Using multiple GCPs, the intersection of the individual projection rays defined by the GCPs and their conjugate image points determine the perspective center, hence the spatial resection label. Ideally, these projection rays intersect in a single point, but this is never the case, and some adjustment method is necessary to relieve the contradictions and calculate the perspective center coordinates. Included in the adjustment are the other camera orientation parameters. When the camera orientation parameters are correct, a 3-D surface model from multiple images may be reconstructed, accomplished by intersecting the projection rays in 3-D space from conjugate points on the multiple images defined by the mathematical model equation for each image. This 3-D model is the basis for one of the mapping products used today: the 1:24,000 USGS (United States Geological Survey) maps. The traditional method of matching model or image points to ground control points is a costly endeavor: buying the material necessary to identify and mark the ground points, make sketches and take notes, then precise measurements of the points using survey crews costs time and resources and everything is done manually. In rural areas where there
may be few natural landmarks, this situation may result in inadequate numbers and a poor distribution of matching points. The human operator evaluating the images then has to find these points on the images. In order to find the control points on the images, the operator needs the sketches and notes about the environment of the point, otherwise the whole image or photo needs to be searched in order to find it.

In computer vision, the determination of the EOPs is known as the “pose estimation” problem. Research in this field is aimed at finding direct solutions to the problem of pose estimation using a minimum of object information. Direct linear solutions for pose estimation are largely based on concepts of algebraic projective geometry. However, the use of projective geometry is not new; homogeneous coordinates, which represent one of its most important concepts, are used for deriving camera parameters as the equivalent to the Direct Linear Transformation common in photogrammetry and remote sensing [60]. Originally proposed by [11], the Direct Linear Transformation (DLT) can be solved without supplying initial approximations for the transformation parameters. The mathematical model of the DLT, derived from the photogrammetric collinearity equation, is a direct linear relationship between image and object coordinates. DLT is a linear treatment of what is essentially a non-linear problem, so it gives only approximate values. However, DLT can be used successfully in work with non-calibrated cameras that do not require rigorous design and estimates of data quality, such as is the case with low cost consumer cameras. DLT consists in first recasting the explicit collinearity equations into linear equations using implicit parameters. Using a sufficient number of control points, six or more for a single image, the implicit parameters are solved via these linear equations. Finally,
the explicit camera parameters can be solved in direct closed-form from the implicit parameters.

For stereo images, the classic orientation is done in two steps: finding Relative Orientation parameters (ROPs) and Absolute Orientation Parameters (AOPs). Absolute orientation (AO) provides parameters that are given with respect to some known reference frame: State Plane or WGS84 system or similar. RO provides parameters with respect to the complementary stereo pair: they are only registered to each other, not a known coordinate system. Relative orientation (RO) is the evaluation of the EOPs of one camera with respect to the camera coordinate system of another camera. ROPs are traditionally determined by finding conjugate points on the images. Using the parameters of this orientation, a scaled 3-D model of the area shown on the images may be created. Traditionally, a stereo pair is first oriented with respect to each other for determination of ROPs, then finding the AOPs, by matching conjugate ground points to the model points, for positioning in some known reference frame. If AOPs are known, the EOPs for each stereo pair can be computed. The difference in EO/RO is whether the image is referenced to a known coordinate system or just to another image. Simultaneous methods for solving this problem, such as bundle adjustments, are now available in a majority of software packages instead of the classic two step method [60]. While theoretically this is easy, it is very difficult to implement, as all the conjugate rays need to intersect to create the surface. Lidar as been of assistance recently to recreate the surface, instead of the traditional stereoscopic method.

As RO can be considered a solved task, although with caveats as expressed in [135], our focus is on the fusing of different modality datasets for finding the EOPs,
although the proposed method works for the RO of images, too. Several commercial systems are available for automating the RO process: HATS by Helava/Leica, Phodis-AT by Zeiss, ISDM by Intergraph-Z/I and MATCH-AT by INPHO [90].

Several interest point detectors are available for finding conjugate points on images: Förstner point operator [53], Moravec point operator [111], [100], and Harris corner detector [67]. These point operators find edges or corners in images and more filtering is necessary to find conjugate points on images. Other methods, like SIFT (Scale Invariant Feature Transform), GLOH (Gradient Location and Orientation Histogram), SURF (Speeded Up Robust Features) and dense matching actually locate the conjugate points and will be described later in Section 2.5.8. These work well with conjugate stereo images, but fail with different data modalities or oblique images. In contrast to this, human operators are remarkably adept in finding conjugate, or identical, features independent of data type [146]. A crucial step in both classic AO and EO is georeferencing: finding and matching conjugate entities. Matching, in general, deals with measuring the similarity of an unlabeled object against labeled objects stored in large image collections. Automated matching tools are commonly used in various domains including mobile mapping [143], wide area surveillance [188], analysis of temporal changes [189] and classification and retrieval from datasets [192].

Whether for stereo or single images, GCPs are widely used for orienting. Alternatively, GCPs can be eliminated by using other data: GIS or lidar datasets georeferenced to the same system as the image data, and automatically extracting interest points/features from these datasets to match interest points/features on the images. A nice property of indirect orientation is that during georeferencing the instrument deficiencies are compensated by the process itself: the errors and deficiencies cancel
out if acquisition and georeferencing of the images is done with the same instrument, since the instrument may contain minor errors. This is not the case with direct orientation: since there is no control, errors due to instrument inaccuracy may affect the results [146].

Photogrammetry, both classic analog and digital, is point-based: control points, Gruber points, interest points, conjugate points and tie points. This makes most photogrammetric applications dependent on the use of distinct points [66]. Point-based methods with experienced personnel are processed well in a traditional setting, but not in the automatic setting of digital photogrammetry. To develop more robust and accurate methods, higher level features like lines, curves, areas, need to be used: free-form linear features have unique properties and are richer than point features [65] [68] and as linear features are more robust, gross errors associated with erroneous point matches are minimized [177]. The vast majority of solutions to the EOP/ROP problem try to explicitly find conjugate points, not complex, general features found in images [62] [54] using some sort of detection operator as listed above or finding distinct points like manholes [42], road junctions [128] or other features in the image for matching. While the majority of works are based on finding conjugate points, there are non-point based methods, but these are also not generic solutions for any situation. Wang et al.’s method [177] only works up to affine transform and requires user interaction to find elongated features, like roads, and this may be considered a limitation of the method. The solution by Lee [94] has the drawback of not producing continuous data, just piece-wise continuous data, and may perform poorly if the assumptions related with the registration/transformation are not satisfied [127]. Many solutions are application dependent or need manual intervention:
creating wireframes [91] or graphs using geometry and topology[176]; in addition, these local registration methods may be sensitive to noise. Habib [65] has a more general solution. This method may use conjugate points or features, however, since it is based on the coplanarity condition, false matches may occur, as any point in the epipolar line is considered conjugate, and further weeding of the data is necessary to find conjugate points. Many of these methods rely on rigid transformations and cannot deal with local deformations adequately. Rigid transformations refer to translation, rotation and scale [49]. A brute force solution, such as generic point matching using dense matching, is now feasible, due to the increased storage and processing capability available [93]. In comparison to these solutions, the proposed solution to the matching problem is applicable to any point distribution and dataset.

Developing automated tools for matching with robust performance on the levels of human cognitive accuracy pose design challenges stemming from several factors including: partial occlusions, viewpoint changes, varying illumination, cluttered backgrounds, intra-category appearance variations, and different types and modalities of sensors. During image acquisition information was lost, as 3-D data was projected to 2-D, and distorted, due to the non-vertical photo axis, or due to occlusions. Matching between the points/features in these various and disparate datasets can be defined as the establishment of correspondences for georeferencing; for example, objects extracted from a digital image are matched against digital data in a GIS. Matching belongs to the class of so-called inverse problems, which are known to be ill-posed. A problem is ill-posed, if no guarantee can be given that a solution exists, is unique, and/or is stable with respect to small variations in the input data. Matching is
ill-posed for various reasons. For instance, for a given point in one image, a corresponding point may not exist due to occlusion, there may be more than one possible match due to repetitive patterns or a semi-transparent object surface, and the solution may be unstable with respect to noise due to poor texture [134]. In addition to these challenges, matching methods are highly affected by the kinds of features extracted from objects, the approaches employed to hypothesize possible matches between two objects, and the conditions for ascertaining the correctness of hypothesized matches [79].

To repeat from the introduction, the purpose of this research is an image registration approach that does not rely on individual points, but linear features: edges and curves. This research presents an alternative, “point-free” approach to solving the matching problem for robust orientation parameter estimation. This approach can be used when entities from two disparate modality datasets are available, whether the data is from discrete lidar points or continuous GIS data. While the approach does solve for explicit correspondence in the end, a good initial approximation is necessary, like uncorrected image perspective center coordinate data and image orientation data from GPS/IMU. The mathematical model is the probability density function of the edgemap error of an image, based on the normal distribution in the 2-D image plane, between two closest edges of the various datasets. Parameter estimation is simultaneously calculated by minimizing the cost function by embedding it in the distance transform of an edgemap of an image. Such an approach estimates the parameters of the mathematical model and implicitly determines the entity matches that correspond to that parameter solution. The benefits of this system is that it provides the co-alignment of the different datasets to a common reference frame, without the
need for explicit conjugacy, although the final result can be used to provide conjugate features. Previous theories for alternative data as a source of control for photogrammetric georeferencing hinged on the ability to identify common items in both datasets. The proposed method provides a practical solution to the elimination of the need for ground control points to establish the georeferencing parameters of the photogrammetric data by utilizing alternative data. For the experiments, a combination of GIS, lidar data and digital images are used, and this input is what the proposed method will use to obtain IO and EO parameters. For the lidar data, it is hypothesized that it has been previously adjusted to existing ground control. The images used were assumed to be refined for refraction and radial distortion, not “raw images”. While the focus is on automating EO and IO, it is shown to be an alternative solution to automatic RO for stereopairs.

In the following sections, a brief background about camera orientation parameters and alternative data to GCPs is presented. The details of the distance transform and its application towards solving the orientation problem are covered in Section 4.3. Experiments with real data are presented in Section 6, followed by conclusions and recommendations for future work.

2.3 Points vs. Features

The traditional method of photogrammetric registration has been point based, requiring matching GCPs (Ground Control Points) to their conjugate image points, which is a costly endeavor that includes buying material necessary to identify and mark the ground points, making sketches, taking notes and measuring of the points using survey crews. This is a time and resource consuming process, all of which is
performed manually. The human operator evaluating the images then has to find these GCPs on the images. In order to find the control points on the images, the operator needs the sketches and notes about the environment of the points, otherwise a high resolution image needs to be searched to find them.

Given the direct orientation solution, GCPs and their matching points in the image plane, accurate camera parameters are estimated using the mathematical models of indirect orientation method, described in section 2.2. Since the direct orientation gives an approximate solution, GCPs should be easier to locate on the image plane.

To move beyond the traditional registration, linear and areal features are more advantageous than points in solving photogrammetric problems such as registration, orientation, matching, reconstruction and recognition: there are abundant linear features available on a given scene especially in an urban environment. In addition, such linear features have semantic information of real world objects on the Earth [147], since they represent discontinuities in boundaries: edge of building, edge of lake, curb on street and street centerline. Similarly, surface features may be part of a building or distinct terrain. Although numerous point features exist, most of them have no semantic meaning in object space, a point, in itself can occur anywhere. Hence, point features used for registration may not contribute much to later stages of photogrammetric processing. On the other hand, linear and surface features used for feature based registration methods can be used for reconstruction and recognition. Various research [147] [63] [64] discusses the advantages of using features in registration procedures. Some significant advantages are the existence of abundant features than points in a given scene, control information in the target coordinate system is
widely available in the form of features and reliability of correspondence in matching is higher for features.

Feature-based matching compares features extracted from the objects in various images, using edge detection or other methods to find matches. The central principle underlying feature modeling is the observation that many objects can be represented by a small number of parts arranged in a characteristic configuration. The match, like strength of an edge or curvature, is measured as a cost function [29]. Newly developed methods, such as probability based Mutual Information (MI) is used for registration via a function optimization [40]. MI has the potential for registering different modality data, like laser and image data.

Besides point matching, to solve the matching problem for matching between datasets, areas have also been suggested [80]. Existing matching methods can be grouped according to area, feature or symbolic based as illustrated in Table 2.1 [143].

<table>
<thead>
<tr>
<th>Method</th>
<th>Measure</th>
<th>Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>correlation, least squares</td>
<td>gray values</td>
</tr>
<tr>
<td>Feature</td>
<td>cost</td>
<td>lines, regions</td>
</tr>
<tr>
<td>Symbolic</td>
<td>cost</td>
<td>other (parameters, graphs...etc.)</td>
</tr>
</tbody>
</table>

Area based matching is associated with gray values. The gray value distribution of two areas, or patches, is compared by correlation or a least squares method. The disadvantage of this method is that it is not flexible with respect to viewpoint changes, although it has been shown to work for normalized images [143, chap. 10]. Normalized image pairs refer to images where the epipolar lines are the scan rows:
the same features appear in the same rows for each image pair. This is similar to template matching, where a known, existing object in the image is searched for a given template that matches the query object.

Besides areas, extracted edges have been used to find RO parameters. A method to compare extracted edges is suggested in [146] by using the tangent along the edge as a function of the distance along the edge, labeled as the $\psi - S$ method, with $\psi$ representing the tangent and $S$ the distance along the edge from a starting point, although this would not work well with high distortions or occlusions. Edges fitted to spline based-methods have been successfully used to find EO parameters [94]. Linear features were also used in [177] to match not just features on images, but from other digital maps. The disadvantages of feature modeling are that it can become very computationally intensive for complex objects, or for objects with many points; along with the inaccuracies from only line correspondences [193]. Symbolic methods derive values from the image or digitized data: moments, wavelets, fractals or other attributes to calculate a similarity measure between the features.

During the last two decades, researchers have been engaged in determining registration methods for free-form curves and surfaces. Iterative Closest Point (ICP), for free-form curves and surfaces, was first introduced by Besl et al. [17]. ICP is used for small transformations [194] where good a approximation of transformation parameters are known or transformation is always gradual. This method iteratively computes parameters by minimizing the distance between a point in the first dataset and the closest point in the second dataset. Due to the ICP algorithm’s flexibility in registering lines, surfaces, triangle sets and curves, it has gained a great deal of attention in the computer vision community, and several variants [139] have been
introduced during last 15 years. Considering the volume of literature, a detailed description of ICP is given in section 2.5.7.

2.4 Registration methods

The proposed method could work with any of these methods, given that there is some approximation of the transformation parameters. Once the parameters are given, the proposed method, along with the transformation equation, can be used to refine the parameters. The proposed method is used in an experiment to demonstrate its capabilities using laser scanning data and aerial images to determine the EOPs of the images using the collinearity equation.

2.4.1 Registration of images

Remotely sensed images have to be transformed into an Earth coordinate system to use in mapping applications. Satellite and aerial images are registered by determining the EOPs of the sensor when the image was acquired.

For satellites, the sensor model and sensor IOPs are not given by the vendor, hence orientation methods are not adopted for satellite images. Other alternative methods, such as the Rational Functional Model (RFM) of polynomial registration are generally used [85]. These methods require GCPs and the number of GCPs required varies as per the order of polynomials used. The detailed description about these methods are elaborated in sections 2.5.5 and 2.5.6. These alternative registration methods are approximate and used only for small scale mapping. Registration of aerial images is an orientation procedure and it can be either direct or indirect. Similarly, terrestrial images also use indirect orientation procedures.
2.4.2 Registration of laser scanning data

Generally, airborne laser scanners have onboard GPS and IMU, thus the position of a laser point is determined similar to direct orientation methodology. The terrestrial laser scanning data are registered to a fixed local coordinate system. It usually has its origin in the sensor’s perspective center [144]. If multiple ground scans are used, then they have to be registered to each other using a 3-D transformation method. Generally, a 3-D rigid body transformation is used for registering multiple scans, done by the Iterative Closest Point (ICP) algorithm that registers lidar data using actual data points or resampled point-point correspondence [139]. Resampling may be necessary, since unlike other registration methods discussed above, there are usually no point to point correspondences between lidar point clouds. The ICP working principle and variants are discussed in section 2.5.7. Airborne laser scanning data and aerial images give complementary information that are essential for reconstruction and recognition of surfaces [9]. Hence, to exploit the advantages of both, it is necessary to co-register them. This co-registration method requires a 3-D similarity transformation. A number of such co-registration methods exist: [146], [61], [59], [93], [183] to name a few.

2.4.3 Registration of maps/GIS

There are abundant useful maps available in paper and digital format. In order to archive them, they have to be scanned and registered using one of the 2-D registration (section 2.5) methods. It is done by identifying points on the map and by entering the coordinates to an Earth reference system manually. Map-to-map registration is

\[\text{http://www.usgs.gov/pubprod/}\]
the process of registering the map in a scanner/pixel coordinate system to the map in an Earth coordinate system. It is done by a choice of a 2-D registration method and common points in both systems.

2.5 Point based registration models

The aim of registration is to determine the geometric relation between two or more geometric representations of the same features. Unlike matching, the point correspondence is always known a priori in registration problems. Conventional registration is done by identifying conjugate points on both data sets and minimizing the sum of squared residuals in X, Y coordinates for 2-D registration and in X, Y, Z coordinates for 3-D registration. The mathematical model for the registration of a 2-D dataset, a 3-D dataset and orientation are given below for comparison [118].

2.5.1 2-D similarity registration

A 2-D conformal or similarity transformation is defined by one scale \( S \), one rotation \( \alpha \), translation in X direction \( tx \) and Y direction \( ty \); and \( rx \) refers to the residual in the X coordinate and \( ry \) refers to the residual in the Y coordinate.

\[
a = S \cos \alpha, \tag{2.1}
\]
\[
b = S \sin \alpha, \tag{2.2}
\]
\[
\begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} rx \\ ry \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}, \tag{2.3}
\]

where \( x', y' \) refer to coordinates of the target system.
Transformation parameters \(a, b, tx\) and \(ty\) are determined by solving equation 2.3. A minimum of two conjugate points are required to use this model. If there are redundant conjugate points available, the least squares solution of the transformation parameters is determined by minimizing the sum of squared residuals [143].

### 2.5.2 Affine transformation

This 2-D transformation method uses different scales \(S_x, S_y\), respectively in the \(x\) and \(y\) directions, along with the skew angle \((\epsilon)\), and the non-orthogonality of the axes. Transformation parameters \(a, b, c, d, tx\) and \(ty\) are determined by solving equation 2.8. A minimum three conjugate points are required to use this model and redundant conjugate points would lead to a least squares solution by minimizing the sum of squared residuals of the target system coordinates [136].

\[
\begin{align*}
a &= S_x \cos \alpha, \\
b &= -S_y \sin(\alpha + \epsilon), \\
c &= S_x \sin \alpha, \\
d &= S_y \cos(\alpha + \epsilon), \\
\end{align*}
\]

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} rx \\ ry \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix},
\]

where \(rx\) refers to the residual in the \(X\) coordinate and \(ry\) refers to the residual in the \(Y\) coordinate, \(x', y'\) refer to coordinates of the target system, \(tx, ty\) refer to translation in the \(X, Y\) axis, and \(\alpha\) refers to the rotation of the axes.
2.5.3 3-D registration

3-D conformal transformation defines the transformation between two 3-D frames by parameters one scale \((S)\), three rotations \(\omega, \phi, \kappa\) respectively around the \(X,Y,Z\) axes, three translations \(tx, ty, tz\). If a point to point correspondence exists between the datasets, then this mathematical model can be used. A minimum of three conjugate points are required to obtain a solution. In case of redundant observations, a least squares solution is used to adjust the data \([143]\).

\[
\begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix}
+ \begin{bmatrix}
x_x \\
y_y \\
z_z
\end{bmatrix}
= SR
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
+ \begin{bmatrix}
tx \\
ty \\
tz
\end{bmatrix},
\]

where

\[
R = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \omega & -\sin \omega \\
0 & \sin \omega & \cos \omega
\end{bmatrix}
\begin{bmatrix}
\cos \phi & 0 & -\sin \phi \\
0 & 1 & 0 \\
\sin \phi & 0 & \cos \phi
\end{bmatrix}
\begin{bmatrix}
\cos \kappa & -\sin \kappa & 0 \\
\sin \kappa & \cos \kappa & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

(2.10)

\[
\hat{R} = \begin{bmatrix}
\cos \phi \cos \kappa & -\cos \phi \sin \kappa & \sin \phi \\
\cos \omega \sin \kappa + \sin \omega \sin \phi \cos \kappa & \cos \omega \cos \kappa - \sin \omega \sin \phi \sin \kappa & -\sin \omega \cos \phi \\
\sin \omega \sin \kappa - \cos \omega \sin \phi \cos \kappa & \sin \omega \cos \kappa + \cos \omega \sin \phi \sin \kappa & \cos \omega \cos \phi
\end{bmatrix}
\]

(2.11)

and \(S\) refers to Scale, \(\omega, \phi, \kappa\) refer to rotation of the coordinate frame with respect to the \(X,Y,Z\) axes respectively, \(R\) is a 3-D rotation matrix consisting of rotations \(\omega, \phi, \kappa, tx, ty, tz\) refer to translation in the \(X,Y,\) and \(Z\) axis, \(X', Y', Z'\) refer to
coordinates of the target system, \( rx \) refers to the residual in the \( X \) coordinate, \( ry \) refers to the residual in the \( Y \) coordinate and \( rz \) refers to the residual in the \( Z \) coordinate.

### 2.5.4 Exterior orientation

Exterior orientation of an image refers to the parameters that define the location and rotation of the camera reference frame with respect to a known world reference frame. The collinearity model is the mathematical model for determining the EOPs of aerial images. It is based on the principle that the perspective center, image point and corresponding object point are collinear. Collinearity equations for point features is given by equation 2.12. Orientation parameters \( X_c, Y_c, Z_c, \omega, \phi, \kappa \) are determined by measuring GCPs and conjugate image points. Collinearity equations for point features is given by eq. (2.12) for a camera. This equation describes a light ray passing through the perspective center from image point \( (x, y) \) to its conjugate ground point \( (X, Y, Z) \), using the exterior orientation parameters and interior orientation parameters:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
X_c \\
Y_c \\
Z_c
\end{bmatrix} + \lambda M \begin{bmatrix}
x - x_0 + rx \\
y - y_0 + ry \\
-c
\end{bmatrix}, \quad (2.12)
\]

where

\[
M = \begin{pmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
r_{31} & r_{32} & r_{33}
\end{pmatrix} \quad (2.13)
\]

\[
= \begin{pmatrix}
cos(\phi)cos(\kappa) & sin(\omega)sin(\phi)cos(\kappa) + cos(\omega)sin(\kappa) & -cos(\omega)sin(\phi)cos(\kappa) + sin(\omega)sin(\kappa) \\
-cos(\phi)sin(\kappa) & -sin(\omega)sin(\phi)sin(\kappa) + cos(\omega)cos(\kappa) & cos(\omega)sin(\phi)sin(\kappa) + sin(\omega)cos(\kappa) \\
sin(\phi) & -sin(\omega)cos(\phi) & cos(\omega)cos(\phi)
\end{pmatrix}
\]
which is the same as in eq.(2.11) described previously; where $x_0, y_0$ are the principal point coordinates, the offset between the image center and the projection of the perspective center to the image plane, $X_c, Y_c, Z_c$ are the coordinates of camera’s perspective center, $\omega, \phi, \kappa$ are the attitude of the camera (rotation angles around $X, Y, Z$ axis of the ground coordinate system, counter clockwise is positive rotation), $x, y$ are image coordinates of the points, $X, Y, Z$ are ground coordinates of image points, $c$ is the calibrated focal length of the camera, $\lambda$ is the scale factor between the model space and object space, $rx$ refers to the residual in the $x$ coordinate, $ry$ refers to the residual in the $y$ coordinate, and $r_{11}...r_{33}$ are the elements of the 3-D orthogonal rotation matrix $M$ containing sine and cosine functions of the three angles eq.(2.14).

This is independent whether the camera is a digital camera or a film-based camera. The parameters of the images acquired by airborne digital cameras are rather well known: the camera model is based on the very same central perspective projection used for large-format aerial film cameras. One of the notable difference to film comes from the radiometric behavior: digital cameras respond to incoming light in the 0.4 – 1.1 micron range, while films are sensitive to a larger range [169]. The other notable difference is that the digital cameras are much smaller. Depending on the optical filter used, digital images may cover only the visual part of the spectrum or some subpart of it. In addition, digital cameras exhibit a linear characteristic and provide a much a finer intensity resolution compared to analog film [146]. By rearranging equation 2.12 and eliminating the scale factor $\lambda$, the two image coordinate equations emerge, eq.(2.14) for the $x$ coordinate, and eq.(2.15) for the $y$ coordinate:

$$x + rx = x_0 - c \frac{(X - X_c)r_{11} + (Y - Y_c)r_{12} + (Z - Z_c)r_{13}}{(X - X_c)r_{31} + (Y - Y_c)r_{32} + (Z - Z_c)r_{33}},$$  

(2.14)
\[ y + ry = y_0 - c \frac{(X - X_c)r_{21} + (Y - Y_c)r_{22} + (Z - Z_c)r_{23}}{(X - X_c)r_{31} + (Y - Y_c)r_{32} + (Z - Z_c)r_{33}}. \] (2.15)

The inverse of this equation, the inverse collinearity equation, is used to calculate the ground coordinates from image coordinates, shown in equation (2.16) and equation (2.17). Since this equation describes the ray traveling from the image plane through the perspective center to the object, the elevation of the object point has to be known in order to obtain coordinates, otherwise the intersection can happen on any surface, as the end point is not defined, Fig. (2.3).

\[
X = X_c + (Z - Z_c) \frac{(x - x_0)r_{11} + (y - y_0)r_{21} - cr_{31}}{(x - x_0)r_{13} + (y - y_0)r_{23} - cr_{33}}, \] (2.16)

\[
Y = Y_c + (Z - Z_c) \frac{(x - x_0)r_{12} + (y - y_0)r_{22} - cr_{32}}{(x - x_0)r_{13} + (y - y_0)r_{23} - cr_{33}}. \] (2.17)

Figure 2.3: The inverse collinearity equation needs a reference elevation.
The collinearity equation thus links the camera orientation parameters from EO and IO, and relates the points on the ground to the points on the image. If the ground points and their conjugate image point coordinates are known, then the camera orientation parameters can be acquired using an iterative least squares solution by linearizing the collinearity equation for the $x$ and $y$ coordinates. As the collinearity equation shows, IO and EO cannot be separated: as the focal length and PPO change during image acquisition, due to temperature, air pressure and platform movement, the EOPs will change accordingly. Thus both IO and EO have to be estimated continually, in order to achieve accurate results. Due to the perfect correlation between camera the focal length and the vertical coordinate, a small difference of about 20 $\mu$m between assumed focal length from lab-calibration and the true focal length during camera exposure, for example, will result in a systematic height offset of about 20 cm for 1:10,000 image scale.

It can be realized from equations 2.14 and 2.15 that at least three ground control points are required for a solution. Redundant points provide a least squares solution of EOPs by minimizing the sum of squared residuals of the image coordinates. For pixel-based calculation, the image coordinates have to be converted to row, column format.

The collinearity equation is the fundamental model used in photogrammetry [182].

### 2.5.5 Polynomial registration

This method registers the satellite image with a set of GCPs and an equation with selected polynomial order. The satellite image is warped to either first order/second order/third order surfaces. The increasing order of the polynomial would
increase the number of GCPs required for registration. GCPs must be distributed all over the image for reliable results. This method is generally opted for only flat areas. An example of the first order polynomial registration formulas are given below.

The coordinates of an image target system and GCPs are related by equation 2.19. The polynomial coefficients \( a_0 \) to \( a_2 \), \( b_0 \) to \( b_2 \) are determined by measuring at least three GCPs (first order) and conjugate image points. Having more than three points would lead to a least squares solution. This also involves image resampling [136], as the pixels have to be transformed to a different coordinate system.

\[
\begin{align*}
    r &= a_0 + a_1 X + a_2 Y, \\
    c &= b_0 + b_1 X + b_2 Y,
\end{align*}
\]  

(2.18)

(2.19)

where \( r, c \) are the row and column of the image and \( X, Y \) are the coordinates of GCPs.

### 2.5.6 Rational functional model, Direct linear transfer

The Rational Functional Model (RFM) is a method of registering satellite images using the ratio of two polynomial functions [164]. This model is used to register images when the sensor model of the satellite camera is not available to the data buyers. Sensor model refers to the parameters that are required to determine the geometry of the captured data. The number of Rational Function Coefficients (RFCs) varies based on the order of polynomial used in the RFM. Some satellite image vendors directly provide RFCs instead of providing sensor parameters. It is determined by generating a set of virtual gridded control points set to the whole image extent, as well as to the elevation. Using these virtual control points, RFCs are determined for
each image and distributed to the clients. If the sensor model is not provided and also RFCs are not provided by the data vendor, it can be determined using a set of GCPs. RFM serves as an alternative model for the actual sensor model. Hence, orthorectification, DEM generation are also done with RFCs [162], [163]. However, RFM is not suitable for urban area where terrain elevation changes drastically. Using a higher order polynomial and huge number of GCPs alone will not be sufficient for accurate registration or DEM generation. This also involves image resampling [136], as the pixels have to be transformed to a different system.

An example RFM with first order polynomials is given in equation 2.21. It relates the rows and columns of the image to the coordinates of the GCPs (X, Y and Z). By measuring sufficient number of GCPs (seven for first order), the RFCs $a_0$ to $a_3$, $b_0$ to $b_3$, $c_0$ to $c_3$ and $d_0$ to $d_3$ are determined.

$$r = \frac{a_0 + a_1 X + a_2 Y + a_3 Z}{b_0 + b_1 X + b_2 Y + b_3 Z}, \quad (2.20)$$

$$c = \frac{c_0 + c_1 X + c_2 Y + c_3 Z}{d_0 + d_1 X + d_2 Y + d_3 Z}, \quad (2.21)$$

$r, c$ are the row and column of the image, $X, Y$ are the coordinates of the GCPs.

Equation 2.21 can be compared with collinearity equations 2.12 given in section 2.5.4. Instead of the total of six independent orientation parameters necessary, 16 are used. It not only increases the number of GCPs required for registration, but also causes numerical instability in the adjustment due to correlation between the parameters [38]. The first order RFM is identical to DLT, described previously above.
2.5.7 ICP(Iterative Closest Point)

The ICP(Iterative Closest Point) algorithm is based on the assumption that the closest points on two different datasets are conjugate. Given the datasets, each dataset is sampled with a number of data points. The distances of each sampled point on one dataset to all points in the second dataset are computed. The closest point in one dataset to each sampled point in the other dataset is considered as conjugate, and based on that two datasets are registered. There are several invariants of ICP algorithms [139] available and only a few of them are discussed here.

Mathematical model

If $P$ and $P'$ represent points from the same surface, the Euclidean distance between any two points with coordinates $P(x, y, z)$ and $P'(x', y', z')$ is given as:

$$d = \| \vec{P} - \vec{P}' \| = \sqrt{(x' - x)^2 + (y' - y)^2 + (z' - z)^2}. \quad (2.22)$$

For each point in set $P$, the conjugate point is the closest point in $P'$.

Similarly, for calculating the distances between a point and line, triangle, or parametric entity, the reader is directed to [17].

The basic idea behind the ICP is that, if correspondences between points are known, then the transformation parameters are acquired by solving for the transformation parameters that minimize the distances between point pairs. In ICP, it is assumed that the closest points are conjugate points. This method requires a close initial approximation of the transformation parameters. From the initial position, conjugate points are determined based on the closest point principle. Transformation
parameters can be determined from such a point to point correspondence and this process is continued until the sum squared distances reaches a minimum. ICP utilizes a rigid transformation model: translation and rotation only.

ICP variants

Rusinkiewicz [139] summarizes different variants of the ICP algorithm. The authors classify the algorithms based on stages of ICP, such as selection of points in one or both datasets, matching of points, weighting the correspondence, rejection criteria and the minimization function. At almost all stages of ICP, people have proposed some variations either to enhance its speed or to make it more robust. The following sections show some of the ICP variants based on the criterion used by [139].

Selection of points can be all data points in the datasets, uniform sampling of both datasets, random sampling of datasets or other methods. Similarly, there are different variants for matching criteria for the closest point: the point on a line/surface with closest orthogonal distance from the point or other selection. There are different ICP methods that use different weights for the points used in the registration parameter determination. The weight could be constant for all point pairs. Also, there are methods that use a higher weight for the smaller closest distance and a smaller weight for the higher closest distance. Some variants of ICP reject the point pairs with distances above a certain threshold, using a certain percentage of this thresholded data. The least squares minimization function minimizes the sum of squared distances. There is a variant of this minimization function that includes the minimization of color/gray value, in addition to minimization of the distance.
ICP issues

Pita [129] lists out major issues with the ICP algorithm. The most important are related to the initial approximation and overlap in two different datasets. If the surface is flat or has no significant features, then ICP would fail. If the data with partial overlap needs to be registered, then it is advisable to use only the overlapping portions of the datasets for ICP registration. Otherwise, ICP will most likely fail. If the initial approximation is not close, then ICP would converge to a local minimum rather than to a global minimum.

Since the development of the ICP registration method, linear features and surfaces are widely used for registration. However, the registration is based on point to point correspondence of linear features. Therefore, the usage of linear features in registration is not completely exploited. This is partially due to the unavailability of mathematical models that handle features as features, instead of points. Schenk [147] used parameterization of linear features for aerial triangulation. Considering the fact that linear features generated from two different sensors or operators or time periods may not be the same, a sophisticated mathematical model is required to handle this difficulty.

Fitzgibbon [52] argues that point-set registration is better performed using a general-purpose nonlinear optimization procedure using the Levenberg-Marquardt optimization than by the ICP algorithm. The general conclusion of this paper is that specialized algorithms such as ICP are not always to be preferred to general-purpose techniques. The general-purpose routine is faster, and much simpler to program. This is in agreement with the proposed research, which is a general method for registration,
instead of concentrating on specialized methods, as the registration methods listed above.

### 2.5.8 Image based point matching

Several image based matching solutions exist in photogrammetry: SIFT (Scale Invariant Feature Transform), GLOH (Gradient Location and Orientation Histogram), SURF (Speeded Up Robust Features) and dense matching.

**SIFT**

SIFT is an algorithm in computer vision to detect and describe local features in images. The algorithm was published by Lowe [99].

SIFT “key points”, or interest points, are first extracted from a set of reference images and stored in a database. A key point is recognized in a new image by individually comparing each point from the new image to this database and finding candidate matching points based on the Euclidean distance of their descriptors. From the full set of matches, subsets of key points that agree on the location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient database table. Each cluster of three or more features that agree on an object and its orientation is then subject to further detailed model verification and subsequently outliers are discarded. Finally, the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence [99]. Matching can be further strengthened by using RANSAC (Random...
Sample Consensus) to further filter out outliers. RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers. It is a non-deterministic algorithm in the sense that it produces a reasonable result only with a certain probability, with this probability increasing as more iterations are allowed [51].

A SIFT descriptor is a 3-D histogram in which two dimensions correspond to image spatial dimensions and the additional dimension to the image gradient direction, normally discretized into eight bins. Each bin contains a weighted sum of the norms of the image gradients around its center, where the weights roughly depend on the distance to the bin center [168].

SIFT can identify objects even among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to scale, orientation, and affine distortion, and partially invariant to illumination changes [100]. It should be noted, after some personal experiments, that although the author claims affine distortion invariance, this distortion has to be minimal, otherwise SIFT will fail. A similar claim can be made for projective invariance: if the projectivity is low, SIFT will work around it, as in the case of near vertical aerial photography. If the distortion is high, SIFT will fail.

**GLOH**

GLOH is a robust image descriptor that can be used in computer vision tasks. It is a SIFT-like descriptor that considers more spatial regions. The higher dimensionality of the descriptor is reduced to 64 through principal components analysis (PCA) [106]. This allows more matches than SIFT, but at the same speed as SIFT.
SURF

SURF is a robust image detector and descriptor. It is partly inspired by the SIFT descriptor. The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT [14].

Dense Matching

This descriptor is inspired from earlier ones, such as SIFT and GLOH, but can be computed much faster, and it is utilized at a pixel level, unlike the previous descriptors, which are at a small region level. Unlike SURF, which can also be computed efficiently at every pixel, it does not introduce new data that degrades the matching performance. As a result, unlike competing methods that require many high-resolution images to produce good reconstructions, the descriptor can compute matches from pairs of low-quality images, such as the ones captured by video streams [168]. Since the number of matches increases, the computing time also increases, especially as large amounts of image data are used. Also, for homogeneous areas, there may be incorrect matches, due to the density of the matches.

2.6 General feature-based registration methods

Acka [2] registers 3-D data using a least squares 3-D surface matching method. This method imitates a 2-D least squares image matching method [4] with some modifications. Instead of the sum of squared gray value differences, the minimization of the sum of squared euclidean distances is used. This method registers the overlapping 3-D data, lidar or reconstructed surface from stereo photos, by defining a template
surface and search surface. The principle of this method is similar to ICP, as both of them minimize the squared sum of euclidean distances. This method uses only a small window, hence computational complexity is not as big as ICP. Habib [61] computes the rotation parameters by taking two random points on a line and eliminating the translation parameters, discussed in section 2.6.4. Similar to the other approach, the translation parameters are determined by substituting the rotation parameters in the 3-D transformation equation. Both these methods require at least two lines to determine the parameters. This approach has some disadvantages, such as the inability to use free-form lines, which is generally the result of edge extraction.

It should be noted that similar to point-based registration methods, feature correspondence is also known in advance for registration. Some more feature based methods are discussed in the sections below.

2.6.1 Iterative Closest Line

Alshawa [7] uses two methods to illustrate the algorithm. These algorithms first determine the rotation matrix and then translation subsequently. The first method is explained in ICP form. The rotation parameters are determined by minimizing the sum of differences of the direction vectors of the line in two datasets. Once the rotation parameters are determined, the translation parameters are determined by taking two random points on a line in one system and expanding the formula corresponding to a 3-D transformation. The significant advantage over ICP is that ICL works on line to line correspondence and not point to point correspondence.
2.6.2 Feature based aerial triangulation

Aerial triangulation is the method of determining the EOPs of multiple overlapping images simultaneously. Schenk [147] discusses a method of using straight lines in aerial triangulation rather than points. The author generates unique representation of straight lines in 3-D space and a mathematical model that does not require point to point correspondence between straight lines. Collinearity equations are changed to accommodate line parameters and the corresponding model is derived. The equation is similar to 2.12, however, instead of conjugate points, a line equation is used [190]:

\[
\begin{align*}
x + rx &= x_0 - \frac{f(X(t) - X_c)r_{11} + (Y(t) - Y_c)r_{12} + (Z(t) - Z_c)r_{13}}{(X(t) - X_c)r_{31} + (Y(t) - Y_c)r_{32} + (Z(t) - Z_c)r_{33}}, \\
y + ry &= y_0 - \frac{f(X(t) - X_c)r_{21} + (Y(t) - Y_c)r_{22} + (Z(t) - Z_c)r_{23}}{(X(t) - X_c)r_{31} + (Y(t) - Y_c)r_{32} + (Z(t) - Z_c)r_{33}},
\end{align*}
\]

(2.23) (2.24)

where \( t \) is a parameter on a line that contains the conjugate point.

The great advantage of this method is that images do not need to overlap, just referenced to the same line on the ground, which can be a street, building or other long, linear feature. Obviously, there are some configurations that do not lead to a solution with straight lines. This method could be extended for any free form linear feature.

2.6.3 Matching surfaces using surface normal

Jaw [83] demonstrates an aerial triangulation method using control surfaces. The author demonstrates the method using a hypothetical plane derived from images and a surface plane as its control surface.
2.6.4 Co-Registration of different modalities using linear features

Habib et al [61] discusses a method of co-registering straight lines from aerial images and airborne laser scanning data. The authors use straight lines from airborne laser scanning data as control information to orient the straight lines created from relatively oriented aerial images. First, a pair of conjugate lines from model and laser scanning data are chosen. This pair does not need to have the same end points. Then by choosing a point on the model line and transforming it into the laser scanning system using equations 2.9, an additional scale parameter is obtained, referring to the location of the point on a line, in addition to the 3-D similarity transformation parameters. For two such points, there are two scale parameters. By eliminating translation and scale parameters the rotation parameters are determined. By substituting the rotation parameters into the point correspondence equation, the scale and translation parameters are determined. This method needs user interaction.

2.6.5 Linear features registration

Wang et al [177] implement registration of linear features in two steps, namely feature extraction and shape matching. In the feature extraction step, linear features are extracted using active contour methods with manual initialization. An active contour is a deformable spline model. The standard solution makes use of several nodes positioned in the vicinity of the linear feature to be extracted and proceeds through an optimization process, in which the manually added nodes move to new locations that minimize the spline function. The function is an expression of optimality criteria for the linear segment to be extracted, and as such, comprises various radiometric
(edge sharpness) and geometric (smoothness, continuity) terms, Fig.(2.4). The critical node points, which correspond to a substantial variation in the road geometry, are extracted from the active contour. The variation in geometry is defined by the local curvature of each discrete node on the contour. Critical nodes on are detected by thresholding the curvature of the spline. In order to retain all convex and concave points, a small threshold is selected.

An active contour is created for both images of a stereo-pair using similar features, and the resulting curvatures at critical node points are compared. Though the authors have claimed to handle the problem of non-existence of point-to-point correspondence, the proposed matching and transformation parameter estimation is totally based on point-to-point correspondence which is a major drawback of the approach. In addition to that this method uses local curvature for selecting critical node points. Curvature is highly sensitive to noise, hence there are chances that critical point could be noise.

Figure 2.4: Active contour example. Black nodes are manually selected nodes on road, white nodes are detected edge points on road via active contour method. The manually selected black nodes were moved by the function to the road edge.
In [159], the concept is that linear features will be extracted from individual photographs or in a stereo model. This is concentrated on the automatic matching of the two linear features using curvature. These features could be obtained by manual or automatic means in an analytical plotter or by using digital techniques. The corresponding feature will be coordinated in three dimensions, by any suitable means: GPS or digital map plus terrain model. The features are then automatically matched to find the “best” fit together. This method is similar to the proposed method, in that there is no explicit point-to-point correspondence. Finally the transformation for three dimensions for absolute orientation or perspective transformation for exterior orientation, will be performed from the corresponding point coordinates of the matched curves.

2.7 Comparison of registration methods

The previous sections give an overview of current registration algorithm available. The oldest and most famous is the point based registration method, which has a simple mathematical model and minimization function. For automation of point based methods, like SIFT, outliers can be removed easily with RANSAC. It was suggested that RANSAC could be used for image analysis, matching, camera calibration, and location determination applications. Rather than using matches with a high percentage of blunders and trying to eliminate invalid matches, RANSAC starts with a small dataset and enlarges the dataset with consistent data whenever possible. This method is quite similar to using the Simultaneous Hough Transform to simultaneously solve for all the involved parameters. There is no concept of partial overlap, which is a major advantage as compared to feature-based methods. Many feature-based algorithms
would fail if feature overlap is partial. If well-defined points exist in two datasets, then point-based registration should be used. Point based registration methods have been used traditionally, since they work well with highly trained personnel. However, automating point-based methods is rather difficult, as exact conjugate point pairs need be found. Choosing conjugate points is difficult, even in an interactive environment.

ICP has significant advantages over other point-based registration methods: the closest point is considered a conjugate point. If initial approximations are good, then this method works well and the versatility of this algorithm is also considered to be an advantage. ICP is capable of using different minimization methods, registering 2-D and 3-D- features and registering line, curve, lidar data and Triangular Irregular Network (TIN) [17]. This method has several disadvantages as well. If non-rigid body transformation is used, this method will fail most likely unless very close approximations are given. In the ICP algorithm and its variants, the main emphasis is put on the estimation of a six-parameter rigid body transformation without uniform scale factor [2]. Though this method claims to use all kinds of features, it uses only point approximation of features, thus it is point to point correspondence. Properties of features, such as shape and orientation, are not exploited. ICP causes problem if partial features from occlusion are used. The dataset that has to be registered must be identical [2]. The parameters of the rigid body transformation are generally estimated by the use of closed-form solutions, such as singular value decomposition (SVD) [8]. The closed-form solutions can estimate only six parameters of a rigid body transformation or seven parameters of a similarity transformation, and cannot fully consider the statistical point error models [194]. The ICP algorithm always converges monotonically to a local minimum with respect to the distance objective.
function. This monotonic convergence behavior leads to slow convergence, which typically means 30-50 iterations [17]. The main computationally expensive part of the ICP is the exhaustive search for the correspondences. Besl and McKay [17] reported that 95% of the run-time is consumed in searching the correspondences. Speeding up the correspondence computation is another option in order to accelerate the ICP.

SIFT, SURF, GLOH are widely used in software solutions for stereo image matching and give good initial approximations [133]. Dense matching has become popular, due to the better solutions than the previously listed solutions, but may take longer. The drawback of all these methods is the computing power necessary.

The major disadvantages of using point based methods are [178] and [64]:

- Point features are not generally used in later stages of photogrammetry or mapping such as reconstruction and recognition. However, these points can contribute to the surface elevation in photogrammetry.

- A line is more well-defined than its end points. End points can change, but line orientation would remain the same.

- Since most of the points are arbitrary and have no semantics, it is less likely that the same point feature would exist in multi-temporal images or even from a different view of the area at the same time.

- Control information from GIS or lidar data are available in the forms of surface, free-form curves.

Non-point based methods include the use of free-form or parametric curves or surfaces [190], [119], [81], [159] and [94]. These methods have the advantage over
points in that no single point needs to be matched or even identified, and thus errors associated with erroneous point matches are minimized, an advantage in areas where there are few known control points. However, conjugate features still need to be identified before adjustment.

Least Squares Surface Matching (LSSM) [2] is the 3-D version of least squares matching [5]. The author transforms the search surface into a first-order surface such as plane (triangulation) or bi-linear (grid). A template surface is formed, this is the surface that is searched for in the entire dataset. The template surface is placed on the search surface. For irregular data, for every point in the template, the three nearest points are chosen in the search space and a plane is formed. The euclidean distance to the plane is determined from these points. Similarly, the euclidean distance is determined for all points in a template and 3-D similarity transformation parameters are determined by minimizing the sum of squared euclidean distances. It should be noted that like ICP, this method also requires good approximations. Also, if the template surface is small, it is essential to define a template surface that is unique, otherwise least squares surface matching may not converge, due to the size of the data. In addition to that, there has to be multiple matches that are distributed over the dataset. LSSM can be faster than ICP, since only a search window is used instead of the whole dataset.
Chapter 3

PROPOSED RESEARCH

Existing registration methods have significant advantages as well as disadvantages.

Advantages

• There are number of feature based registration methods proposed with line to line correspondence [147, 61, 65] with efficient mathematical models, which do not require point-to-point correspondences.

• ICP has proved to be efficient when the transformation is small or initial approximations are good [194].

• Registration of a dense point cloud generally works with methods that use point to point correspondence. It is due to the reason that dense point clouds most likely have conjugate points. Even if outliers exist, then the outliers can be removed efficiently.

• Classical point based 2-D registration methods discussed in sections 2.5.1, 2.5.2 are linear and initial approximations are not required.
Disadvantages

- Though there are many registration methods available, still there are no complete free-form curve to free-form curve registration or free-form surface to free-form surface registration available. Either the methods use point to point correspondence, although do not need to be conjugate, or point to plane correspondence.

- Features from two sensor/epoch datasets are different samples of the same real-world object. There are still no registration methods that take this fact into account.

- Most of the existing feature-based registration methods require very close initial approximations and explicit conjugacy, making data fusion of different modalities difficult.

It can be inferred from chapter 2 and above summary that registration/fusion is not a solved problem especially when it comes to feature based registration across the modalities. Hence, it is essential to develop a method that takes advantages of existing methods as well as overcome the stated shortcomings. Considering all of these conditions, a more complete “point-free” based registration method is proposed for free-form curve registration. The method for free-from curve registration determines the registration parameters by using the distance function between corresponding free-form curves in two coordinate systems. This method does not require point to point correspondence, parts of the curve can be missing and there is no explicit search for conjugacy. Also, previous correspondence is not necessary and it does not require the point on a linear feature to lie on its conjugate linear feature, which makes the
proposed algorithm completely unique and the error distribution does not have to follow any assumption made for the data, and the error distribution model can be any function.

There is no guarantee that data from different sensors are similar, or even data from the same sensor from different times is the same, as it is affected by several internal, external and user factors. Internal factors would include systematic errors, sensor type, sensor resolution, view angle and spectral bandwidth. Some external factors are illumination, flying height and time of data acquisition. Linear features from images are generally extracted by a human operator or an edge extraction algorithm. If free-form linear features are extracted automatically, then user factors are choice of edge operator, threshold, edge linking algorithm and feature extraction methods. If they are extracted by human, user factors are interpretation skill, quality of work, expertise and even the mood of the operator.

3.1 Point-free method

Recently, researchers have reported good results in the automated fusion of lidar and image data [183],[93]. For instance, Ultracam X is using automated aerial triangulation [59],[93] for mapping and DEM creation. The commonality of these methods is that they all use control points, or tie points to match images. This matching produces a dense network of control and tie points, but it comes at a large computing cost.

Instead of using conjugate points, a feature-based approach is utilized for robust orientation parameter estimation in this research. This approach can be used when entities from two disparate modality datasets are available, the data can be in the
form of images, discrete lidar points or continuous GIS data. While the approach does not require explicit correspondence, a good initial approximation of the transformation parameters is necessary, like uncorrected image perspective center coordinate data and image orientation data from GPS/IMU. The mathematical model is the probability distribution function of any distribution in the 2-D image plane, between the two closest edges of the datasets, and parameter estimation is simultaneously calculated by minimizing the sum of distances. There are two general methods of parameter estimation. They are least-squares estimation (LSE) and maximum likelihood estimation (MLE). The former has been a popular choice of model fitting in photogrammetry and is tied to many familiar statistical concepts such as linear regression, sum of squares error and root mean squared deviation. LSE, which unlike MLE requires no or minimal distributional assumptions, is useful for obtaining a descriptive measure for the purpose of summarizing observed data, but it has no basis for testing hypotheses or constructing confidence intervals [117] [86]. An excellent comparison of the two methods can be found in [117]. The maximum likelihood method is conjectured to be less sensitive to outliers, as the cost function is based on the probability distribution function of the edge detection error function. The probability distribution function of the error in the implementation is theorized to fit the normal distribution, but it can be any distribution function.

The purpose of this research is to minimize the novel fitting error using both LSE and MLE for comparison, by embedding the cost function in the distance transform space. The distance transform provides the probability of each image edge pixel being a GIS or lidar edge pixel, projected onto the image, based on the distance between the image edge pixel and the projected GIS or lidar edge pixel. Since the projected GIS or
lidar pixel location depends on the camera orientation parameters, an optimization function may be used to find the minimal cost function, which, in turn, results in the desired parameters. The distance transform is a powerful feature space, where the structures of interest are considered in a higher dimension and represented using signed Euclidean distance transforms [127]. Given observation data and a likelihood function, MLE is the method of estimating the parameters of the likelihood function that create the largest value for that function from the observed data, thus giving the maximum likelihood estimate [103]. According to [117, 86, 122], when observations are independent of one another and are normally distributed with a constant variance, MLE is equivalent to minimization of the sum of squared error, and therefore, the same parameter values are obtained under either MLE or LSE. The MLE method will be used with the constraint that all edges are of equal weight, there will be no prior information on the edges, except that they are buildings or curbs. A further investigation can be made into giving the edges different weights based on some prior information or gradient magnitude, like building edges are stronger than sidewalk edges, or even based on the initial GPS/IMU location, edges could be given a different weight if the edges are extracted from an urban area or a rural area.

3.1.1 Distance Transform

The distance transform (DT) is a general operator forming the basis of many methods, with great potential for practical applications [18]. The distance transform is based on the distance function. The mathematical definition of the distance function can be found in [56]; it is a function which defines a distance between elements. The distance transform is an operator normally only applied to binary images. The
distance transformation converts a binary digital image, consisting of feature and non-feature pixels, into an image where all non-feature pixels have a value corresponding to the distance to the nearest feature pixel [20], fig(3.1)a and b. There are several distance functions: Euclidean distance, chessboard or Chebyshev distance, where the distance is the greatest of their differences along any coordinate dimension and the Manhattan distance, where the distance between two points is the sum of the (absolute) differences of their coordinates. Other distance transform functions are the Hausdorff distance [105], geodesic distance and chamfer distance [30]. The Euclidean distance function for the distance transform is used in the study experiments, since it is the most intuitive, although computationally the chamfer distance is faster. For detail on the different distance metrics and distance transform algorithms, the reader is directed to a comprehensive survey in [35].

While the Euclidean distance function itself is well understood, solutions to utilize this function on images focus on the algorithms used [50] [48], due to speed of computation considerations. For the application, the distance transform is used since it allows for a non-specific search after edge detection of the image or the projection of lidar data; the distance between the two datasets is already embedded in the transform itself, and thus no specific, point-to-point matching search is necessary.

Both the EO and RO parameter calculation will utilize the DT, along with IO. In both cases, edge detection will create a set of edges in stereo images. In the case of EOP calculation, the extracted building edges from the lidar data will be used for a distance-based comparison to the edge detection on the image. The extracted laser edges will be projected onto the images using the collinearity equation, thus these points are a function of the EOPs. The closest points between the two edge datasets
Figure 3.1: (a) Distance transform, $D$, computed from a closed edge segment superimposed as white line. Larger distances have lighter shade. (b) Euclidean distances computed from a central pixel that has a binary value of 1 in a 5x5 neighborhood. Image (a) represents the actual image, (b) represents the values of the distance transform. The left side of (b) represents an edge, where the edge pixels are represented as a binary value of 1, non-edge pixels are shown as 0. The right side of (b) shows the result of the distance transform, where the numbers represent the closest distance to the edge points on the left side.

are found via the distance function, and a least squares adjustment is applied, using the initial EOPs as the first approximation to minimize the sum of distance residuals. The distance function, by default, provides the shortest distance between the two edge datasets; in addition, it provides for the coordinates of the closest points. The updated EOPs are used for projecting the laser point on to the images, and this continues until
convergence. In the case of ROP calculation, the relative displacement of the stereo pair will be used to find conjugate points. From the conjugate point calculation, the ROP can be calculated.

3.1.2 Applications

Distance transforms have been considered either as a feature [87] or as an optimization space [92] for image processing operations, such as computing the skeleton, morphological dilation and contraction, construction of shortest paths between points, and shape factor computation image registration/segmentation. This section presents of few applications of distance transformations.

The distance function has been in use in computing for over several decades. As early as 1968, used distance functions on digital images [138]. One of the first notable uses was for embedding a surface for level set-based methods [154, 123]. Borgefors [18, 19] proposes to use distance transformations for pattern matching. Paglieroni [125] studies the correlation properties of the Euclidean DT. Applications are shown for pattern matching and for stereopsis. In traditional stereopsis, conjugate points are found in the images. The distance between the locations of the same point in both images reveals its depth, i.e., its distance from the camera. The conjugate of a point in the right image is constrained, by camera orientation parameters and geometry, to lie on a line in the left image: the epipolar line. Its exact position must be determined by contextual information, like edge matching: application of an edge detector on both images, then computing the Euclidean DT from both edge images.

The matching criterion is a composition of the correlation of an edge image with the DT of the other, and vice-versa. Toivanen [167] [166] uses the distance transform for
route optimization. More recent utilizations have been for skeletonization of images [141, 33, 31], image enhancement [191, 16, 151, 184], and flow simulations [27].

3.1.3 Comparison of LSE and MLE

Least squares estimates are calculated by fitting a regression curve to the points in a probability plot. Maximum likelihood estimates are calculated by maximizing the likelihood function. The likelihood function describes for each set of distribution parameters the chance that the true distribution has these parameters based on the sample. MLE uses a probabilistic framework, unlike LSE.

Here are the major advantages of each method [1]:

**Least squares**

- The probability plot has a better graphical display because the curve is fitted to the points.

- For small samples, LSE is more accurate than MLE. MLE tends to overestimate the shape parameter for a Weibull distribution and underestimate the scale parameter in other distributions.

The probability distribution function of a Weibull random variable $x$ is:

$$f(x; \lambda, k) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0, \\ 0 & x < 0, \end{cases}$$

where $k > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution [126].

**Maximum likelihood**

- For heavily filtered samples, MLE is more accurate than LSE.
• Distribution parameter estimates are more precise than LSE.

• MLE enables analyses performance.

Both MLE and LSE have attractive mathematical qualities. Specifically, [142, 1],

• They become minimum variance unbiased estimators as the sample size increases. By unbiased, meaning that if taking an increasingly large number of random samples with replacement from a population, the average value of the parameter estimates will be theoretically exactly equal to the population value. By minimum variance, meaning that the estimator has the smallest variance, and thus the narrowest confidence interval of all estimators of that type

• For large samples, the approximate normal distributions and approximate sample variances can be used to generate confidence bounds and hypothesis tests for the parameters

The advantage of the method of maximum likelihood is that all of the information in the data is used, rather than just the first and second moments, as is the case with least squares. Another advantage is that many large sample results are known under very general conditions. One disadvantage is that specifically the joint probability distribution function of the process must be used [34].

LSE, even though the process gives good results, is sensitive to outliers and not robust to variations, such as occlusion boundaries. It provides no information about the significant uncertainty of match or possibility that a qualitatively incorrect position has been found. MLE provides uncertainty estimation, a robust measure and better results for greyscale and edge templates [186], [115].
When possible, both methods should be tried. If the results are consistent, then there is more support for the conclusions.

3.2 Assumptions

Assumptions made about the proposed methodology:

- The proposed method does not require matches; it, however, requires initial approximation from GPS/IMU for EO or SIFT for RO for the two datasets, and the final result is explicit match detection,

- This adjustment method is non-linear, hence approximate values of the transformation parameters are required in advance,

- The edge detection from the images contains errors, these errors are modeled by a Gaussian function [137], by superimposing the edges with the 2-D Gaussian. This will represent the uncertainty of the actual edges,

- The lidar data is adjusted and correctly georeferenced, and serves as ground control; the lidar errors will not be taken into account at this time, although, like the image edge detection uncertainty, they could be modeled by a 2-D Gaussian function

- Image refinement, section 4.1.3, is assumed to be performed.

3.3 Significance

The proposed method has several significant advantages:
• No point to point correspondence is used, hence there is no need for interpolation or resampling of points.

• The number of points and features in the two datasets can be different and the shape can be projected differently, or parts of it missing/occluded.

• Even partially overlapping free-form curves can be registered.

• Without any further processing, this feature-based registration will provide a partial reconstruction of surfaces, as well as partial extraction of GIS features, such as roads and buildings for GIS data update.

### 3.4 Some Properties of Estimators

For an estimator to be useful, it has to be accurate and precise. LSE fits this description [142], as does MLE [1], and this is why they were chosen for this research.

Let the random variable \( X \) have a pdf, probability distribution function, that is of known functional form, but in which the pdf depends on an unknown parameter \( \theta \) that may take any value in a set \( \Theta \), where \( \Theta \) is the actual parameter. The pdf can be written as \( f(x; \theta), \theta \in \Theta \). To each value \( \theta \in \Theta \) there corresponds one set of value or values. To find precisely the \( \theta \) that describes the pdf of the random variable, a point estimate of \( \theta \) is necessary.

The estimate of \( \theta \) by some function of the observations \( X_1, \ldots, X_n \) is called a statistic or an estimator. A particular value of an estimator, \( t(x_1, \ldots, x_n) \), is called an estimate. An estimator is itself a random variable, so its behavior for different random samples will be described by a probability distribution.
A “good” estimator needs to be somehow “centered” with respect to $\theta$. If it is not, the estimator will tend either to under-estimate or over-estimate $\theta$. A further property that a good estimator should possess is precision, that is, the dispersion of the distribution should be small. These two properties need to be considered together. It is not very helpful to have an estimator with small variance if it is “centered” far from $\theta$. The difference between an estimator $T = t(X_1, \ldots, X_n)$ and $\theta$ is referred to as an error, and the “mean squared error” defined below is a commonly used measure of performance of an estimator [116].

### 3.4.1 Unbiasedness

For a random sample $X_1, \ldots, X_n$ from $f(x; \theta)$, a statistic $T = t(X_1, \ldots, X_n)$ is an unbiased estimator of $\theta$ if $E(T) = \theta$. The bias in $T$ (as an estimator or $\theta$) is [142]:

$$b_T(\theta) = E(T) - \theta.$$  \hspace{1cm} (3.1)

It is the distance between the average of the collection of estimates, and the single parameter being estimated. It also is the expected value of the error, since:

$$E(T) - \theta = E(T - \theta).$$  \hspace{1cm} (3.2)

A relatively high absolute value for the bias means the average position of the estimates is off-target, and a relatively low absolute bias means the average position of the estimates is on target. They may be dispersed or may be clustered. The relationship between bias and variance is analogous to the relationship between accuracy and precision.
3.4.2 Mean Square Error (MSE)

For a random sample $X_1,\ldots,X_n$ from $f(x;\theta)$ and a statistic $T = t(X_1,\ldots,X_n)$, which is an estimator of $\theta$, the mean square error (MSE) is defined as:

$$MSE = E[(T - \theta)^2].$$  \hfill (3.3)

The mean squared error is defined as the expected value, probability-weighted average, over all samples, of the squared errors; that is,

$$MSE(T) = E[(T(X) - \theta)^2].$$  \hfill (3.4)

It is used to indicate how far, on average, the collection of estimates are from the parameter being estimated. For example, even if all estimates are the same, yet grossly miss the parameter, the MSE is still relatively large. Note, however, that if the MSE is relatively low, then the estimates are likely more highly clustered than highly dispersed.

For a given sample $x$, the sampling deviation of the estimator $T$ is defined as

$$d(x) = T(x) - E(T(X)) = T(x) - E(T),$$  \hfill (3.5)

where $E(T(X))$ is the expected value of the estimator. Note that the sampling deviation, $d$, depends not only on the estimator, but on the sample.

The variance of $T$ is simply the expected value of the squared sampling deviations, that is,

$$Var(T) = E[(T - E(T))^2].$$  \hfill (3.6)
It is used to indicate how far, on average, the collection of estimates are from the expected value of the estimates. Note the difference between MSE and variance. A relatively high variance means the estimates are dispersed, and a relatively low variance means the estimates are clustered. Some things to note: even if the variance is low, the cluster of estimates may still be far off-target, and even if the variance is high, the diffuse collection of estimates may still be unbiased. Finally, note that even if all estimates grossly miss the expected value, if they nevertheless all have the same value, the variance is zero.

The MSE can be expressed alternatively as [142],

\[
E[(T - \theta)^2] = E[(T - E(T)) + (E(T) - \theta)]^2
= E[(T - E(T))^2] + [E(T) - \theta]^2.
\]

Thus,

\[
\text{MSE} = \text{Var}(T) + b_T^2(\theta). \hspace{1cm} (3.7)
\]

Now from (3.7), the MSE cannot usually be made equal to zero. It will only be small when both \( \text{Var}(T) \) and the bias in \( T \) are small. So rather than use unbiasedness and minimum variance to characterize “goodness” of a point estimator, the mean square error may be employed.

Considering the problem of the choice of estimator of \( \sigma^2 \), of a sample, based on a random sample of size \( n \) from a \( N(\mu, \sigma^2) \) normal distribution, \( S^2 = \sum_{i=1}^{n} (X_i - \)
\( \bar{X}^2/(n - 1) \) is often called the sample variance and has the properties [116]

\[
E(S^2) = \sigma^2, \quad \text{(so } S^2 \text{ is unbiased)}
\]

\[
\text{Var}(S^2) = 2\sigma^4/(n - 1).
\]

The maximum likelihood estimate of \( \sigma^2 \), \( \sum_{i=1}^{n}(X_i - \bar{X})^2/n \), will be denoted by \( \hat{\sigma}^2 \).

Now \( \hat{\sigma}^2 = \frac{n-1}{n} S^2 \) and

\[
E(\hat{\sigma}^2) = \frac{n - 1}{n} \sigma^2 = \left( 1 - \frac{1}{n} \right) \sigma^2
\]

\[
\text{Var}(\hat{\sigma}^2) = \frac{(n - 1)^2}{n^2} \text{Var}(S^2) = \frac{2(n - 1)}{n^2} \sigma^4.
\]

Thus, \( \hat{\sigma}^2 \) is biased. To calculate \( \hat{\sigma}^2 \) it is necessary to extract the mean, consuming one degree of freedom.

Now \( \hat{\sigma}^2 \) is biased, and evaluating its mean square error using the identities [142]:

\[
E\{E(Y|X)\} = E(Y)
\]

\[
\text{var}\{E(Y|X)\} \leq \text{var}(Y)
\]

\[
\text{var}(Y) = E\{\text{var}(Y|X)\} + \text{var}\{E(Y|X)\}
\]
\[ E\{\text{var}(Y|X)\} = E\left[E(Y^2|X) - E(Y|X)^2\right] \]
\[ = E\left[E(Y^2|X)\right] - E\left[E(Y|X)^2\right] \]
\[ = E(Y^2) - E\left[E(Y|X)^2\right] + [E(Y)]^2 - [E(Y)]^2 \]
\[ = \left[E(Y^2) - E(Y)^2\right] - E\left[E(Y|X)^2\right] + E\left[E(Y|X)\right]^2 \]
\[ = \text{var}(Y) - \text{var}\{E(Y|X)\} . \]

Therefore,
\[ \text{var}(Y) = E[\text{var}(Y|X)] + \text{var}[E(Y|X)] . \]

Then using the above identities [116]

\[
\text{MSE } \hat{\sigma}^2 = \frac{2(n-1)\sigma^4}{n^2} + \left[\sigma^2 - (1 - \frac{1}{n})\sigma^2\right]^2 = \frac{\sigma^4(2n-1)}{n^2}.
\]

Now for \( S^2 \),

\[
\text{MSE } S^2 = \text{Var}(S^2) = \frac{2\sigma^4}{n-1} > \left(\frac{2n-1}{n^2}\right)\sigma^4 \quad (3.8)
\]

since \( \frac{2}{n-1} > \frac{2n-1}{n^2} \) for \( n \) an integer greater than 1. So for the normal distribution, the MLE of \( \sigma^2 \) is better in the sense of MSE than the sample variance.

### 3.4.3 Consistency

A further desirable property of estimators is that of consistency, which is an asymptotic property. To understand consistency, it is necessary to think of \( T \) as really being \( T_n \), the \( n^{th} \) member of an infinite sequence of estimators, \( T_1, \ldots, T_n \).
Roughly speaking, an estimator is consistent if, as \( n \) gets large, the probability that \( T_n \) lies arbitrarily close to the parameter being estimated becomes itself arbitrarily close to 1. More formally [116],

\[
T_n = t(X_1, \ldots, X_n) \text{ is a consistent estimator of } \theta \text{ if }
\lim_{n \to \infty} P(|T_n - \theta| \geq \epsilon) = 0 \text{ for any } \epsilon > 0.
\]

(3.9)

This is often referred to as convergence in probability of \( T_n \) to \( \theta \).

An equivalent definition, for cases where the second moment exists, is

\[
T_n = t(X_1, \ldots, X_n) \text{ is a consistent estimator of } \theta \text{ if }
\lim_{n \to \infty} E[(T_n - \theta)^2] = 0.
\]

(3.10)

That is, the MSE of \( T_n \) as an estimator of \( \theta \), decreases to zero as more and more observations are incorporated into its composition.

Asymptote ideally defines a curve coming close to a line, without actually becoming the same as that line, like the function \( \frac{1}{x} \) coming arbitrarily close to the \( x \) axis at infinity, when \( x \to \infty \), without actually becoming the same as the \( x \) axis. So as the sample size increases, \( T_n \) approximates closer to the true value. When \( n \to \infty \), the sample is the entire population. If \( T_n \) is not convergent, it would not be a consistent estimator.

### 3.4.4 Efficiency

The term is frequently used in comparison of two estimators where a measure of relative efficiency is used. In particular, given two unbiased estimators, \( T_1 \) and \( T_2 \) of
\( \theta \), the efficiency of \( T_1 \) relative to \( T_2 \) is defined to be [116]

\[
e(T_1, T_2) = \frac{\text{Var}(T_2)}{\text{Var}(T_1)},
\]

(3.11)

and \( T_2 \) is more efficient than \( T_1 \) if \( \text{Var}(T_2) < \text{Var}(T_1) \).

Note that it is only reasonable to compare estimators on the basis of variance if they are both unbiased. To allow for cases where this is not so, using MSE in the description of efficiency:

an estimator \( T_2 \) of \( \theta \) is more efficient than \( T_1 \) if [116]:

\[
\text{MSE} \, T_2 \leq \text{MSE} \, T_1,
\]

(3.12)

with strict inequality for some \( \theta \). Also the relative efficiency of \( T_1 \) with respect to \( T_2 \) is

\[
e(T_1, T_2) = \frac{\text{MSE} \, T_2}{\text{MSE} \, T_1} = \frac{E[(T_2 - \theta)^2]}{E[(T_1 - \theta)^2]},
\]

(3.13)
Chapter 4

BACKGROUND FOR IMPLEMENTATION OF METHOD

Whether point-based or non-point-based, the registration methodology has to tackle the basic registration procedure components, mainly: registration features, mathematical function, and similarity assessment. However, the main questions that rise are: How could a matching procedure between various, disparate datasets be applied, to achieve registration? Lidar data has irregular spacing; and a GIS is, usually, vector data. Images are represented by pixels, discrete areal samples of the surface. How could two datasets, which are not distributed equally and may not have conjugate points, hence lacking one-to-one correlation, be co-registered? The developed method is not data specific, not sensitive to viewpoint changes, and is limited by the number of objects in the image. It can be used to determine the orientation parameters, whether exterior or relative. Perhaps the one drawback is the number of points involved, but even on a standard Windows XP computer, several thousand points could be used. The requirement to use the cost function is a good initial approximation. This cost function is shown in equation (4.23). It is based on the normal distribution [137], but it could be other functions, if other edge detection methods are used. The normal distribution is a probability density function (pdf).
The pdf of a random variable is a function that describes the relative likelihood for this random variable to occur at a given point in the observation space. A random variable is defined as a real- or complex-valued function of some random event, and is fully characterized by its probability distribution [158]. The pdf of the normal distribution is often used to describe, at least approximately, any variable that tends to cluster around the mean: for a normal distribution the relative likelihood that the mean value occurs is the highest, and the greater the difference of the variable is from the mean, the relative likelihood is lower. All normal distributions are symmetric and have bell-shaped density function curves with a single peak, which is the average value of the variables.

### 4.1 Registration parameters

Given two sets of points in $\mathbb{R}^k$, which for convenience is denoted by *image pixels* and *ground data*, with their elements being denoted by $\mathbf{p}_{i=1}^{N_p}$ and $\mathbf{g}_{i=1}^{N_g}$. The ground data is considered to be adjusted, so that the camera orientation parameters will have to be adjusted that the image pixels are registered to the projected ground data. Edges from the GIS and lidar data and image data are extracted. Edges are theorized to segment the image and delineate objects, so that an edge is the limit of a feature of an object; in a GIS, this is already the case most of the time. GIS layers represent similar objects, like buildings, curbs and roads. The theory is that some of these edges in the datasets are conjugate, others are just labeled as noise, and the aim is to discern the conjugate edges from the noisy data. Although the edges are conjugate, the search is not for explicit conjugate points on the edges. The projection of the ground data to the images is done by utilizing the collinearity equation: the ground data is projected...
on to the image using the initial EOPs and IOPs from the GPS/IMU measurements and camera calibration. Thus, there are nine parameters that need to be adjusted during this process: \( \omega, \phi, \kappa \), for orienting the camera axis at the instant the image was taken and the three coordinates of the perspective center, \( X_c, Y_c, Z_c \), along with \( f \), the camera focal length, and \( x_0, y_0 \), the principal point coordinates, which are the projection of the perspective center onto the image plane, ideally located at the centroid of the image. While focal length and principal point are calibrated for the cameras, they can slightly change due to air pressure, temperature or the handling of the camera. The parameter vector is then: \( \mathbf{a} = [\omega, \phi, \kappa, X_c, Y_c, Z_c, f, x_0, y_0] \). The radial distortion and refraction parameters are considered to be solved or eliminated.

The task of registration is to determine the parameters of a transformation \( T \), which when applied to the data points, best aligns image pixels and projected ground data. By adjusting the EOPs and IOPs, the ground points will be projected to different places on the image. The parameters of \( T \) are represented by the collinearity equation, eq (2.12), repeated here for convenience for the \( x \) and \( y \) coordinates:

\[
x = x_0 - c \frac{(X - X_c)r_{11} + (Y - Y_c)r_{12} + (Z - Z_c)r_{13}}{(X - X_c)r_{31} + (Y - Y_c)r_{32} + (Z - Z_c)r_{33}},
\]

\[
y = y_0 - c \frac{(X - X_c)r_{21} + (Y - Y_c)r_{22} + (Z - Z_c)r_{23}}{(X - X_c)r_{31} + (Y - Y_c)r_{32} + (Z - Z_c)r_{33}}.
\]

Fig.(4.1) shows the process flow diagram.

Alignment is measured by an error function \( \epsilon(|e|) \), where \( e \) is the error, and a typical choice is to define \( \epsilon(e^2) \). For point based methods using varying modality data, the ICP (Iterative Closest Point) algorithm was considered one of the most popular methods for many years \[17\]. The ICP algorithm finds the best correspondence
Figure 4.1: Process flow diagram of the proposed method for exterior orientation between two point sets by iteratively determining the translation and rotation parameters of a 2-D/3-D rigid body transformation. ICP has been used to register targets to lidar scans [13] [180]. However, the ICP algorithm assumes that one point set is a subset of the other which may not be applicable between laser and photogrammetric points, where there may be no conjugate points occurring; this is circumvented by minimizing the area between two free form lines in 2-D, or minimizing the volume between 3-D points [119].

Unlike ICP, the distance transform does not require conjugate points. ICP is mainly used for rigid transformations, and may not give accurate results for other
transformations [119]. Rigid transformation in this case means translation and rotation, with no scale involved. ICP is a widely used method to register lidar datasets to each other and other datasets [153] [12].

The proposed concept is similar to ICP, in that it minimizes distances, but has several key differences. These are:

1. No initial point-to-point correspondence is necessary
2. ICP is a closed form solution
3. ICP works on a rigid transformation model
4. The distance function is continuous, uses the whole shape, in fact, uses clones of the shape - the distance function at various distances
5. The distance function is an implicit function, and has more flexibility in shape representation
6. The distance function not only takes each point into account, but measures discrepancy for all points
7. It is a stochastic model, as opposed to a deterministic model

The proposed method is similar to methods from Sester, [153], which uses affine ICP to register areal data to each other or Habib [64] which uses linear features found in both images and lidar data, however, manual identification of planar patches is done beforehand to ease extraction. Paragios [124], which uses a non-rigid registration method using the distance transform to register shapes to each other, but this method only registers images to each other. The proposed method’s advantages to
the previous methods is that it can handle projective changes along with recovering the IO and EO parameters without any manual intervention, for any dataset.

4.1.1 **Exterior orientation**

A basic geometric problem in photogrammetry is the determination of the EOPs correctly and quickly. This can be made by spatial resection for single images or relative and absolute orientation or bundle adjustment for a stereo model using two or more images. Theoretically, a conventional block adjustment, not supported by airborne GPS, requires a minimum of two horizontal and three vertical ground control points [5]. A block is the images of two or more side lapping strips used to cover an area [182]. Ideally the more ground control points in a project the better, as their huge number ensures better geometric stability of the block and high redundancy. In reality, photogrammetric companies try to minimize the number of ground control points because of the high cost of field operations [97]. On the other hand, GCPs have to provide a sufficient number of them to ensure acceptable accuracy of EOPs. Acceptable accuracy means that it is in the same range as would result from a fully controlled individual model [3]. The requirement of minimal forward overlap is 60% and it guarantees a common area on three consecutive photos. All tie points measured in this area are considered strong, as they notably minimize error propagation. Tie points are photogrammetric control points that are selected such that they are common to adjacent images, and they are not ground control points as the coordinates are not known, but are only common points in several images. As the name suggest, they tie images together to orient and locate the images relative to each other. The accuracy, number, geometry and distribution of tie point observations
greatly influence the final results of the EOPs [71]. Measuring tie points takes time even for experienced personnel, and this limits the number of extracted tie points. Digital photogrammetry now provides for the automated extraction of a great number of tie points, using software listed previously. The benefits of the automation of tie point selection and measurement are not limited to their increased numbers, better distribution and very accurate measurement techniques, but ensures stronger connections between the images through the huge number of multiple tie points that did not exist with manual extraction. It is important to stress that in a project a large number of these tie points fall on four, five, and even six images. They tighten connections not only between the photos in a strip, but also between the photos in neighboring strips and thus reduce the number of GCPs necessary [97]. A typical side overlap is smaller than 50%, with a minimum requirement of 20%, and as such it does not offer triple overlap between the strips. Therefore the standard requirements regarding location and consequently the number of GCPs in a project not assisted by airborne GPS or any other additional sensor, characterized by 60% forward overlap and 20 to 40% side overlap, are as follows [97]:

- There should be a full (X,Y,Z) control point in every corner of the block and every four to six base lengths along the block’s edge strips.

- There should be a vertical (Z) control point in every strip inside the block, four to six bases apart.

- For the blocks with more than 50% side overlap the requirements are different, as it connects three consecutive strips. The change does not apply to control
points along the edge strips, but all the other control points can be spaced every four to six base lengths in the strip, and four to six strips apart.

- A single strip project requires two full ground control points every four to six base lengths.

A baselength is the distance between two consecutive image perspective centers. The expensive and time consuming GCP measurement can be reduced further by a common bundle block adjustment with perspective center locations available from GPS positioning. This method of combined adjustment is today a standard solution, but it is economical only for larger blocks and it requires a small number of control points, the measurement of tie points and a satisfying block configuration from additional flight lines [77].

With a combination of GPS/IMU, the perspective center position and the orientation angles can be determined in-flight for each image. This gives a wide range of flexibility, for example, in coastal regions, forest and deserts where the traditional tie points of images fail, due to lack of features. The accuracy and also the reliability of direct sensor orientation depends upon the relation of the IMU to the camera, the so-called boresight misalignment [77], fig. (4.2). This has to be determined, and is an additional requirement influencing the economic aspects of the project. For the determination of the boresight misalignment and the final EOPs, GCPs are required.

Eliminating GCPs is the final step in the process. Alternative data, like lidar data or GIS data have a great advantage over traditional GCPs: they are evenly distributed in the image and the same points appear in multiple images. These alternative GCPs then have to be matched to the appropriate points on the images. Of course, for
areas with a lack of features this method will still be wanting, but in urban areas with many distinct features, it is very helpful.

4.1.2 Interior orientation

The science of precise measurement using optical instruments was developed long before the first computer vision or digital photogrammetry based measuring systems became available [104]. A major part of photogrammetric work was performed manually and a high-quality optical instrument was a prerequisite for accurate measurements. These instruments are today referred to as “metric” in contrast to “non-metric” or consumer products, which now dominate everyday life: cell phone cameras and pocket cameras. Expensive metric cameras usually incorporate complex optics, which include calibration reports correcting for various errors. Calibration of these high-quality cameras is performed using highly specialized equipment. System calibration is of vital importance for accurate determination of georeferencing based on the combined use of GPS/IMU data. GPS/IMU calibration will be assumed done for
GPS/IMU data. For a film camera, calibration is given by the calibration report that defines the precision characteristics of IO [10]:

1. calibrated focal length of the camera lens
2. radial and decentering lens distortion characteristics and their coefficients
3. point of lens symmetry
4. lens resolving power
5. filter, shutter, and magazine/platen information
6. position of the principal point with respect to the fiducial marks
7. relative positions and distances between and among the fiducial marks
8. stereo model flatness of the system
9. film distortion parameters

For a digital camera, the film based corrections do not apply. Because of the assumption that images or photos from the cameras are based on central projections, it is of paramount interest to know the locations of the projection center of a photogrammetric camera with respect to the GPS/IMU instrument. A central projection, or pinhole, camera is a simple camera without a lens and with a single small aperture — effectively a light-proof box with a small hole in one side. Light from a scene passes through this single point and projects an inverted image on the opposite side of the box. The human eye in bright light acts similarly, as do cameras using small apertures. Up to a certain point, the smaller the hole, the sharper the image, but the
dimmer the projected image. At first, this question may be trivial: the perspective center of an ideal pinhole camera is clearly in the center of the hole. However, the lens system of a camera is very complex, made up of a collection of lenses. The system has to focus the divergent bundle of rays reflected from a distant object to the image plane to achieve a sharp, bright, unblurred image. The effect of a single lens can be computed from the lens equation from optics. The image formed by a system with several lenses is obtained by repeating the process for a single lens sequentially for all lens elements. Since several lenses are used in a camera, the perspective center location determined during the calibration procedure is a mathematical definition. By definition, the optical center of a lens is the point where the optical axis passing through the lens intersects the image plane of the camera. Alternatively, the optical center is the image point where no distortions appear, radial or tangential [15]. Distortion is of paramount interest whenever the image is used for measurements, since it deforms the image from the calculated errorless position. The radial distortion moves a point away or towards the point of symmetry from its calculated errorless position. This can be divided into symmetric and asymmetric components. Mounting errors of the lens elements is called tangential distortion or decentering error. According to the different categories of aberrations, the tangential distortion is an imperfection and not a distortion, thus decentering error is a more appropriate term [110]. Tangential lens distortion is caused by an eccentricity of individual lenses, amongst a lens compound used in cameras, with respect to the optical axis. Tangential and asymmetric radial distortion is difficult to simulate, and is usually quite small, due to high precision manufacturing [110] [104]. Tangential lens distortion along with photo non-flatness may not be always calibrated, or the frequent change in the calibrated focal
length even investigated by camera manufacturers [110]. Symmetric radial distortion is small, in the range of micrometers, but nevertheless fully documented in camera calibration reports [104]. For film cameras, photo non-flatness is caused by the camera inability to acquire a photo on an exactly flat film and due to deformations that take place during film development or at the time of exposure. Even though they are not always calibrated, frequent changes of camera focal length principal point offset occur due to a change of air temperature and air pressure during flight, which results in a deformation in the image bundle of rays. The complex lens system is also vulnerable to changes during flight due to temperature, air pressure changes, and is inherently vulnerable to distortion, as no lens can be polished perfectly, although distortions are minimized. These small deviations tend to sum up causing errors in the coordinates and change from one photo flight to the other and even from one image to the other. Consequently, these errors should be estimated for each single exposure. In this manner, EO/RO parameters should not be estimated without estimating IOPs at the same time for each image. Besides the mentioned parameters, film cameras also must provide the coordinates for the fiducial marks. The fiducial marks appear on the photos and register the film to the camera frame, once the film is removed from the case, and since the film is handled and developed, the original size of the film will change.

Since coordinates of the camera/image coordinate system are based on measurements from the image, and the perspective center is used in relation to GPS/IMU measurements, the location of the projection of the perspective center onto the image plane must be known. Ideally, this is in the center of the image plane, but due to
inaccuracies, there is an offset. The location of the perspective center projection is known as the principal point. This location is also vulnerable to change.

Both the camera/image and ground/known coordinate system is a right-handed system. In order for the camera/image coordinate system to be a right handed system, the z values of the image points are always the negative of the focal length, $-c$, and the coordinates of the perspective center are $(x_0, y_0, 0)$, where $(x_0, y_0)$ are the principal point offset coordinates measured on the image plane. Ideally, the principal point is projected to the centroid of the image plane, the offsets measure the disparity in $x$ and $y$ directions from the centroid.

As part of the mathematical model, the collinearity equation is used, since it is the fundamental model used in photogrammetry that relates image coordinates to the ground coordinates [182]. It is based on the principle that perspective center, image point and corresponding ground point are collinear: they fall on the same 3-D line. The collinearity equation was described in section (2.5.4).

### 4.1.3 Image refinement

For this paper, the image refinement will be considered accomplished, and a brief description of the errors is presented for overview. The purpose of image refinement is to apply “corrections” to the computed photocoordinates to satisfy the collinearity model which is based on the assumption of an ideal pinhole camera, free of systematic errors, lens, fiducial marks, and film, and straight line propagation through the atmosphere. The effect of image refinement is to ‘move’ the physical location of an image point to a virtual location. That is where the image point would have been obtained by an error free data acquisition system and a homogeneous atmosphere.
Correction for radial distortion

There are two major types of symmetrical radial distortion, as shown on fig.(4.3). When image points get displaced from the errorless location to the position closer to the optical axis, barrel distortion occurs. Alternatively, image points can get displaced to the position further away from the optical axis, in this case pincushion distortion occurs. Barrel distortion is common in wide angle lenses and it therefore dominates the distortion-related research [104].

The symmetrical radial distortion is symmetrical and therefore only a function of the distance, \( r \), from the principle point, identical to the point of best symmetry. The
The effect of radial distortion is conveniently considered by using a polynomial, usually including only odd number exponentials, for example:

\[ \Delta r = a_1 + a_2 r^3 + a_3 r^5 + ... \]  \hspace{1cm} (4.3)

where \( \Delta r \) is the radial distortion value at a distance \( r \) from the principle point. The maximum number of polynomial coefficients should not exceed five. The radial distortion of modern cameras is rather small, a few microns, and “smooth”, hence three coefficients are sufficient to approximate the distortion values well [146]. It has been reported in the literature that the \( a_1 \) coefficient is the most significant distortion inherent in digital cameras [152].

The corrections for a point \( P \) are obtained by:

\[ \Delta r_x = \Delta r \frac{x_p}{r} \quad \Delta r_y = \Delta r \frac{y_p}{r} \]  \hspace{1cm} (4.4)

where \( x_p, y_p \) are the photo-coordinates of point \( P \), \( r = \sqrt{x_p^2 + y_p^2} \), and \( \Delta r_x, \Delta r_y \) the coordinate corrections. The radial distortion \( \Delta r \) is obtained from Eq.(4.3). By convention a positive distortion displaces a point outwardly, modeled by the pincushion distortion. Hence, the refined coordinates are:

\[ x_p^{rad} = x_p - \Delta r_x \quad y_p^{rad} = y_p - \Delta r_y \]  \hspace{1cm} (4.5)

**Correction for refraction**

While refraction is not a calibration issue, it does distort the image. Light is refracted at the interface of different media according to Snell’s law. This media is
the different temperature layers in the atmosphere and the lens system of the camera. Refraction always moves the point outward. The displacement can be modeled by [146]:

\[
\Delta r = K \left( r + \frac{r^3}{c^2} \right)
\] (4.6)

where

\[
K = \left( \frac{2410H}{H^2 - 6H + 250} - \frac{2410h^2}{(h^2 - 6h + 250)H} \right) 10^{-6}
\] (4.7)

\(\Delta r\) is the refraction distortion value at distance \(r\) from the principal point, \(c\) is the calibrated focal length, \(H\) is the flight elevation in kilometers and \(h\) is the elevation of the object point, also in kilometers. Similar to radial distortion, the corrections for a point \(P\) are obtained by:

\[
\Delta r_x = \Delta r \frac{x_p}{r} \quad \Delta r_y = \Delta r \frac{y_p}{r}
\] (4.8)

where \(x_p, y_p\) are the photo-coordinates of point \(P\), \(r = \sqrt{x_p^2 + y_p^2}\), and \(\Delta r_x, \Delta r_y\) the coordinate corrections. The refraction distortion \(\Delta r\) is obtained from Eq. (4.6). This always displaces a point outwardly. Hence, the refined coordinates are:

\[
x_p^{ref} = x_p - \Delta r_x \quad y_p^{ref} = y_p - \Delta r_y
\] (4.9)

4.2 Alternative data

To move beyond the traditional calculation of camera parameters, datasets used may be the previously mentioned laser and radar data, which are range data, GPS
survey data or GIS data. Automatically finding conjugate points for georeferencing in the various modality datasets presents many challenges, since besides being able to find conjugate points, the system has to be user friendly. As expressed in [69], for an orientation program to be practical and usable, it should be:

1. autonomous (as opposed to the term “automatic”, “autonomous” implies that no user interaction whatsoever is acceptable)

2. faster than manual image orientation

3. more accurate than manual image orientation

4. flexible with respect to image and camera type (close range, aerial, satellite images)

5. flexible with respect to different types of control information (points, lines, areas, Digital Elevation model (DEM), laser, GIS.)

6. robust: the module should also work with data of poor quality

7. reliable: the module should have a self-diagnostic procedure by which the success and failure of the computations can be assessed

Based on the above criteria, a cost function that compares the datasets has to be designed to be able to handle different situations without difficulty. This is quite difficult, as there is no guarantee that every attribute of the same object appears in the various datasets. Lidar and GPS survey data is a discrete sampling of the surface, although a GPS survey can locate specific objects, like sewer covers, curbs and centerlines. GIS data is usually orthographic, like a map, and has no side views.
of buildings, hills or other tall objects. Neither dataset contains as much visual information as images. A short background on the different data types is presented.

### 4.2.1 Laser range data

First used by NASA in the mid-1980s, laser ranging, a.k.a. lidar has become an essential complement to photogrammetry for mapping and analyzing a vast range of surfaces. Lidar is also referred to as laser ranging, laser altimetry, laser scanning and LADAR (LAser Detection And Ranging). There are basically two types of lidar systems: pulsed laser and waveform. The pulsed laser system is the predominant form used for topographic mapping. A discrete signal is transmitted from the laser and one or more return signals are recorded. Waveform systems use a continuous signal and a continuous, or nearly so, signal is received. Lidar is a system, which integrates three basic data collection tools: laser scanner, GPS, and IMU unit (along with a cooling system for the scanner). It is, theoretically, not restricted to daylight nor cloud cover like aerial imaging, although if aerial imagery is being collected simultaneously, as it is commonly done, then those limitations will affect the particular project. Laser ranging systems do not create a continuous map of the sensed area, rather the surface is sampled by point data, which thus includes noise, or points that interfere with identification [85]. Lidar started as a topographic tool on large scale projects, such as flood plain mapping, but in the past few years it has gained wide acceptance throughout the geospatial industry for a myriad of projects, including 3-D urban modeling, feature extraction, tree identification and volumetrics, mapping bare earth under thick canopy, road delineation, autonomous vehicles, mobile mapping, and carbon inventory. Ground-based laser ranging is becoming as common as aerial applications [101].
Lidar data points are high density, discrete points, although there is a spread factor for each point as the laser ray is not perfectly coherent and there is a slight divergence from a straight line during data collection due to refraction. The spread of a single point is higher, as the altitude of the platform carrying the lidar system increases. In laser technology, “coherent light” denotes a light source that produces (emits) light of in-step waves of identical frequency, phase and polarization. The laser’s beam of coherent light differentiates it from light sources that emit incoherent light beams, of random phase varying with time and position [70]. These points are the result of a laser pulse traveling from the point of discharge to a reflective surface and then back to a detecting sensor, which measures the time difference from discharge to return; the time multiplied by speed of light is the traveled range. In this regard, lidar data and photogrammetric systems are receiving major attention due to their complementary characteristics and potential [101]. An appealing feature of lidar data is the direct acquisition of numerical 3-D coordinates of object space points. The drawback is that discrete and positional nature of lidar datasets makes it difficult to derive surface semantic information, e.g., surface discontinuities and types of observed structures. In addition to that, reconstructed surfaces from lidar data possess no inherent redundancy and as the pulse rate of laser range sensors continues to increase rapidly, it produces huge datasets. In aerial systems, lidar data points are usually computed in the GPS reference frame, WGS84 for example, and consequently, it makes sense to reference the aerial images to the same system [169]. The sampling density is not only a function of the system, but also depends on the flying speed, pulse rate, scan angle and flying height [181]. In contrast to lidar data surfaces, reconstructed surfaces from photogrammetric measurements possess rich semantic information that can be
easily identified in the captured imagery. Moreover, reconstructed surfaces tend to be very accurate due to the inherent redundancy associated with photogrammetry. The drawback of photogrammetric surface reconstruction is the previously mentioned significant time consumed by the process of manually identifying conjugate points in overlapping images and the marking of GCPs. On the other hand, automating the matching problem remains an unreliable task, especially when dealing with large scale imagery over urban areas. Range data thus may assist in the reconstruction of the 3-D structure of a scene.

The raw lidar data represent ranges with respect to the data acquisition platform/aircraft. After the reconstruction of the aircraft motion and applying some mapping frame, the observation spots are available as a function of the horizontal location, forming 3-D point clusters or lines with the point density depending on flying height and speed, surface slope, sampling frequency and the laser’s field of view. The fact that laser systems provide 3-D coordinates can be considered, on one hand, as their limitation, as virtually no object information is provided. In essence, laser scanning is not capable of any direct pointing to a particular object, and the resulting coordinates refer to the footprints of the laser beam. From a radiometric point of view, the laser system is a narrow-band active sensor, providing a spectral signature of the imaged objects.

The processing of laser data, the related algorithms and their use has been discussed in the mapping community [149] [148]. Of special interest are the removal of spurious spots, the separation of multiple return signals, the comparison of randomly spaced surfaces and the extraction of manufactured objects. An interesting and very essential problem is the error budget of the cleaned laser data, especially that there is
no redundancy in the system, not to mention the extrapolation character of the direct platform orientation [150]. Despite the underlying objective of determining surface data, and thus addressing the overall surface data quality, the emphasis is typically on the accuracy of the individual laser points.

Despite the challenges, lidar data and other data is now being routinely fused with data from other sensors, especially multi-spectral and hyper-spectral cameras [101]. Matching discrete lidar points to image points in itself is problematic even for humans, as lidar data is not continuous, and the points are random, with little chance of hitting a pre-identified control point, or well-defined point, like a building corner. An excellent review of various point detector methods and algorithms, along with their analysis can be found in [133].

Integration of point clouds has also been mentioned in research regarding registration [130] [109] [131]. This entails the previously discussed registration problem between images and lidar data using multiple lidar sources. In order to provide a high density laser point dataset over an extended surface area, many parallel overlapping laser strips are needed. This registration of sequential laser strips has been a major area of research [24] [102].

The average accuracy of a large area lidar surveys are around 0.5 ft (15 cm) horizontal and 0.22 ft (6 cm) vertical [6]. For specific surveys, these values can be more or less.

**Surface, areal, linear data**

When there is no or little control in an area, like Mars or rural areas, using the surface as a control may be necessary. On Mars there are only few precisely
known points that can serve as classical GCPs, but laser range measurements of
the surface are available along with images of the surface [46]. Aerial triangulation
can be performed using the control surface rather than the control points, thus no
identification of measured surface points with respect to their object truth is necessary
by fitting first order planes to the ground data [81]. Determining EOPs using DEM
as control has been described in [44] and by [45]. Schenk [148] has suggested
using control surfaces and minimizing the surface normals from one set of points to
the control surface. Habib [63] extended this by using multiple triangular patches,
finding smooth surfaces by data filtering. Using a DEM to find smooth irregular areas
has also been suggested [80].

Parametric and natural curves were used in [190], [119] and [94] to try to avoid
one-to-one point correspondence. Graphs of roads are used for matching using junc-
tions [176] [57] [128]. Graph matching searches directly for the best object-to-object
mapping that satisfies all local constraints. Extracting roads using active contours,
followed by the construction of a polygonal template upon which a road-to-road
matching process was applied by [177]. This last procedure may not always use
exact point matching by using linear features, specifically roads, as registration ob-
jects. However this method uses local curvature for selecting critical node points,
which is highly sensitive to noise, hence there are chances that a critical point could
be noisy, along with the fact that the active contours on images need user interaction.
Buildings, roads and markings and signs and trees are dominant classes of objects
typically observed in urban scenes. Other classes include pedestrians and cars. Au-
tonomously extracting objects from datasets would simplify many image processing
methods, liberating the user from manual methods, but the listed methods still require user intervention. Roads are an important part of urban scenes. Lidar may be used to scan for flat, drivable surface areas, along with radar for texture of the driving surface. Once identified, the surface is assumed to be road, and can be checked by the digital image from a camera: based on edge detection and segmentation, the entire field of view of the camera is interpreted for features and objects, and potential road areas may be identified. In urban areas, roads are typically black asphalt. Other objects by the roadside can also be identified: curbs, phone poles, traffic lights, signs, fire hydrants and newspaper vending machines, depending on the resolution.

Finding smooth patches in lidar data [63] [12] [82] [150] and smooth rooftops [109] [89] restricts the data allowed for use, since only smooth patches are utilized, and may need user interaction to find these patches. The patch is characterized by the known three dimensional coordinates in the patch center that also serves as a control point. These methods are either application specific or very sensitive to viewpoint change. Extracting features and patches is still not an autonomous process. The boundaries of surface patches are determined by surface discontinuities: abrupt changes in surface orientation, boundary between sensed and occluded surfaces; these may be detected by investigating tangent and gradient changes. While planar surface patches are abundant in scenes that contain many urban scenes, this is very restrictive when examining topography or other objects [148]. Abrupt discontinuities can represent smaller objects, like trees or pedestrians.

Delara [37] carried out a study relating to the utilization of aerial lidar data as GCPs in the block adjustment of aerial imagery. A raster image was generated from the intensity of the aerial lidar data by interpolation, this raster image is compared
with the aerial image to extract distinct points, like the corner points of buildings and utilized as GCPs. Even so, this method has the disadvantage of the difficult extraction of the location of the corner points of the buildings from the raster image generated from the aerial lidar data, not to mention the error involved in the process of interpolating the lidar point data to a grid. Habib [64] uses linear features found in both images and lidar data, and the advantage is that no conjugate points need to be found: the endpoints representing the lines need not be conjugate. The similarity measure is based on 3-D straight lines by finding, through least squares adjustment, the translation, rotation and scale needed to register the datasets to each other. This method is limited only to straight lines.

The 3-D edges of 3-D wireframe are used as control in [91]. Constructive Solid Geometry (CSG) is widely used for computer aided drafting and design (CADD). CADD describes the process of drafting with computer software; it provides the user with input tools for the purpose of creating plans, drafts, plats and designs that can be later used for printing or manufacturing. With this technique an object is composed by taking unions and intersections of several primitive shapes, such as rectangular boxes, spheres, cylinders, cones, and tetrahedra. Such models are described by specifying for each primitive the values of the shape parameters and the six EOPs. Often the EOPs are specified for one primitive only and the parameters of the other primitives are described relative to the first primitive. Boundary representations (B-rep) of objects describe the geometry of the points, edge lines, and surfaces of the object boundaries together with the topological relationships between these points, lines, and surfaces. An object can therefore be described by the coordinates of the corner points and the point numbers that make up the lines and faces [175]. These points are then matched
to image points. Thus, each edge is described by means of 3-D points recorded in a
database. The proposed algorithm in [91] consists of the automatic extraction of con-
trol points in the image followed by a matching stage and a spatial resection, which
aims to compute the exterior orientation parameters. While [91] did not focus on a
high degree of automation for the measurements of a few specific models, it pursued
the development of efficient tools for the measurement of the large class of arbitrarily
shaped parameterized models.

Umeda [172] proposes a method for the registration of lidar and image data, based
on the range intensity of lidar data that is simultaneously acquired with a range image.
It is theorized that the intensities of range and image data correspond to the same
feature, thus instead of edge detection and edge matching, intensities are matched.

4.2.2 GIS data

A traditional, hard copy map is a graphic representation of the Earth, or other
celestial bodies, and its features to scale, and their relationship to each other. While
there is no standard definition of a GIS, it is a decision support system designed to
collect, edit, manage, analyze and display data referenced by spatial or geographic
coordinates. It is understood to include the hardware, software, procedures, data
and personnel required to administer the system [121]. Different GISs may generate
slightly different results for a similar application using the exact same data, due to
different software algorithms or hardware rounding error. GIS data represents real ob-
jects, like a map, such as roads, land use, ground elevation, buildings, with digital map
data usually in the form of some basic graphic features: points, lines and polygons,
although raster images can be overlaid on this data, if they are georeferenced to each
other. Many GISs import directly CAD datasets, using the data that engineers and surveyors collected and planned. The data usually includes meta data: data about data. Meta data may be the projection, coordinate system, collection date, collection method, contour interval, credits and notes. GIS data is from processed surveys: topographic surveys, photogrammetric surveys, GPS surveys and thus may have been adjusted, although for smaller datasets, this is not necessarily true. For example, the Franklin County, Ohio auditor’s office GIS encompasses the entire county, with data ranging from hydrant locations to centerlines to swimming pools; each object having \((x, y)\) Ohio State Plane (South) coordinate data [107], and spot elevation data. For smaller counties, the positioning may not be as precise and accurate as for Franklin County. A GIS dataset has to be up to date to be compared to current images or lidar data. There may be also subtle distinctions in a GIS that may be difficult to observe on an image: road type and dominant tree types in a forest area are stored as attribute data. The attribute data describes the location based coordinate data, a characteristic unique to GIS. The attribute data models may exist internally within the GIS software or may be reflected in external commercial Database Management Software (DBMS). There are six fundamental components of GIS data [21]:

- lineage (where the data is from, what transformations occurred)
- positional accuracy (spatial data)
- attribute accuracy (properties of the data)
- logical consistency (topology)
- completeness (data quality report)
• temporal accuracy (currency)

Each of these has an uncertainty component: the user may not know the origin of the original data, how it was collected, what methods were used and when it was collected; what was the original purpose and topology considerations and how and where it was verified. The user has to keep this in mind when using GIS data for control.

The Franklin County Ohio Auditor’s office states that [108]: “All maps adhere to National Map Accuracy Standards at the original compilation scale of 1" = 100’ with 2’ contours based on North American Vertical Datum 1929 (when displayed) and 500’ grid ticks based on the Ohio State Plane Coordinate System, South Zone and North American Datum 1983”. The National Map Accuracy Standards (NMAS) are a set of standards used to represent confirmation to map compilation standards. It is a USGS document [173] for map accuracy in the United States that “Defines accuracy standards for published maps, including horizontal and vertical accuracy, accuracy testing method, accuracy labeling on published maps, labeling when a map is an enlargement of another map, and basic information for map construction as to latitude and longitude boundaries”. Another good Ohio GIS is the Cincinnati Area GIS (CAGIS) or the Delaware County GIS called Delaware Appraisal Land Information System (DALIS). The City of Dublin, Ohio, GIS website states [43]: “The City of Dublin provides the data within these pages for your personal use “as is.” This information is derived from multiple sources which may, in part, not be current, be outside the control of the City of Dublin, and may be of dubious accuracy. The areas depicted by these maps are approximate, and are not necessarily accurate to surveying or engineering standards. The City of Dublin makes no warranty or guaranty as to
the content, accuracy, timeliness, or completeness of any of the data provided, and assumes no legal responsibility for the information contained on this map.” This is similar to San Bernardino County [140]: “Disclaimer and Limitation of Liability: The geographic information available by and through San Bernardino County is presented as a public resource of general information. The information may include both map data and data provided by the County Assessor. Data provided by the San Bernardino County Assessor’s Office is maintained for internal use only in the determination of property value for the purpose of taxation. While the Assessor’s Office strives to maintain the accuracy of the content of its data files, it makes no claims, promises, or guarantees about the accuracy, completeness, or adequacy of the contents of the files. The County of San Bernardino assumes no responsibility arising from use of the information provided. No warranty of any kind, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for particular purposes is made. It is the responsibility of the recipient of this data to determine that the level of accuracy meets the needs of their application prior to making any judgments or decisions based on this information. Do not make any business decisions based on this data before validating your decision with the appropriate County office. Certain GIS information may not be placed on the Internet as it may result in a violation of Government Code 6254.21.” Generally, the GIS data, regarding the six fundamental components listed previously, of areas of larger populations are quite good in the United States, and smaller populations may be dubious. This can be demonstrated by the GIS features of Franklin, Delaware, and Fairfield counties in Ohio align almost perfectly: roads, boundaries, hydrology and parcels. All these counties contain or are close to the city of Columbus boundaries, with Franklin being
the central county and the rest around and adjacent to Franklin County. Columbus is the largest city in Ohio [28], with most of the city area in Franklin county, but some areas in the surrounding counties. However, a much smaller population county south and adjacent to Franklin, Pickaway county, has the GIS data slightly rotated with respect to the GIS data of the previously mentioned counties; the datasets do not align perfectly. Pickaway is a rural county with a small population. The county contains only small areas of Columbus and its population and thus the tax database is much smaller than the rest of the counties surrounding Columbus.

GIS has been used in EOP determination by [155]. The approach is based on image matching between an aerial image and its corresponding orthoimage in a GIS database, not explicitly GIS data itself.

4.3 Mathematical model of cost function

The proposed method considers the distance from each edge, or data, pixel in the image to the closest matching projected ground point in the image. It is not restricted to overlapping pixels, the edge pixel is not necessarily an explicitly conjugate pixel to the projected ground point. Ground points could be GIS data, which already delineates objects, and thus provides edge data; or it can be edge detected lidar points.

The basic idea of this method is to use a multiresolution matching method that compares edges and projected lidar points based on a coarse-to-fine strategy. Since image edge extraction depends on many variables, and the initial, lidar edge points projected from the initial GPS/IMU EOPs via the collinearity equation, may not be very accurate or precise, longer, “coarse” edges are extracted from the image first,
since these are theorized to belong to buildings, as opposed to shorter edges, which may represent cars or vegetation. Edge refinement will occur in an iterative process, as shorter edges are included in the solution as the EOPs become more accurate and precise. If shorter edges are included in the initial adjustment, the solution may include these edges in the initial match, and give an incorrect solution.

The estimation of the camera parameters is done with the parameters embedded into a function. This function is from the edge extracted lidar points or GIS data projected onto the image edgemap using the initial GPS/INS parameters. Similar to methods based on the Hausdorff distance [74], the distance from each edge extracted lidar point projected to the image edgemap is used. These distances are denoted $d_1...d_n$. In general, these distances can be found quickly for any data, if the distance transforms of the image are pre computed [125]. The joint probability density function (pdf) for the distances, given the parameters, can be approximated as the product of each individual pdf for each point, if the distance measurements are independent [122].

The concept is similar for RO, except instead of projecting lidar points, the ground points will be approximated by computing them from initial ROPs. The initial ground points and the ROPs will be then used in a similar manner as described above. The ROPs will be the result of traditional dependent relative orientation. The parameters maximize this cost function and this results in an estimate for the camera parameters.

### 4.3.1 Data model

Common strategies to model data and use statistical inference to interpret the models include [116]:
Each technique has its strengths and limitations. Bayesian probabilistic formulation for image matching yields several benefits. Perhaps most importantly, an arbitrary likelihood function for the matching error between edge or intensity features can be considered. In this case, a normal distribution is chosen as the pdf. In contrast to frequentist least squares adjustment, Bayesian allows the elimination of the sharp distinction between matched and unmatched points [122]. Frequentist least squares adjustment has a long history in photogrammetry, and it is well understood [104] [182].

The proposed research, in contrast to previous image matching applications, matches between pixels that do not directly overlap are allowed and in traditional photogrammetry may not be considered as strictly conjugate. This allows for a more robust treatment of noise and errors, since no initial correspondences are necessary, hence there is no dealing with explicit conjugacy: finding initial matches using conjugate points. A strategy is described that generalizes previous work for detecting the position that optimizes a variation on the Hausdorff distance.

**Frequentist inference**

The object of statistics is to make an inference about a population based on information contained in a sample. Populations are characterized by parameters, so many statistical investigations involve inferences about one or more parameters. The process of performing repetitions of an experiment and gathering data is called sampling [116].
The basic ideas of random sampling, presentation of data by way of density or probability functions, and a statistic as a function of the data, are assumed known. Computation of a statistic from a set of observations constitutes a reduction of the data, where there are \( n \) samples from a total of \( N \), to a single number: the statistic itself. In the process of such reduction, some information about the population may be lost. Hopefully, the statistic used is chosen so that the information lost is not relevant to the problem. The notion of sufficiency deals with this idea.

Commonly used statistics are: sample mean, sample variance, sample median, sample range. These are random variables with probability distributions dependent on the original distribution from which the sample was taken.

The frequentist least squares adjustment method [55]:

- has properties that are well understood
- is calibrated: properties correspond to long-run frequency
- may be non-optimal due to choice of math function
- does not depend on error distribution
- is useful in applications, such as adjustment calculation for surveying, mapping, photogrammetry or geodesy

**Bayesian inference**

The Bayesian interpretation of probability can be seen as an extension of logic that enables reasoning with uncertain statements[84]. To evaluate the probability of a hypothesis, the Bayesian inference specifies some prior probability from some prior information or data, which is then updated in the light of new relevant data.
The Bayesian interpretation provides a standard set of procedures and formulas to perform this calculation. Bayesian probability interprets the concept of probability as “a measure of a state of knowledge”[84], in contrast to interpreting it as a frequency or a “propensity” of some phenomenon.

The Bayesian probabilistic method [55]:

- has a well defined basis for inference
- is internally consistent
- leads to optimal results from one point of view
- requires a probability distribution: a prior distribution
- is not necessarily calibrated
- is not always clear on how much answers depend on the choice of prior knowledge about the data
- requires high-dimensional integration to create results
- is useful in applications: pattern recognition, city modelling, for example

**Bayesian model**

Let $f(y|w)$ denote the probability density function (pdf) that specifies the probability of observing data vector $y$ given the parameter $w$. Given a set of parameter values, the corresponding pdf will show that some data are more probable than other data. Accordingly, the inverse problem: Given the observed data and a model of interest, find the one pdf, among all the probability densities that the model prescribes,
that is most likely to have produced the data. To solve this inverse problem, the likelihood function reverses the roles of the data vector \( y \) and the parameter vector \( w \) in \( f(y|w) \):

\[
L(w|y) = f(y|w).
\]

(4.10)

Thus \( L(w|y) \) represents the likelihood of the parameter \( w \) given the observed data \( y \) and as such is a function of \( w \) [117]. Given that the different parameter values have different probability distributions, the parameter value of interest is the one that corresponds to the desired probability distribution.

Let \( x \) be an edge point in a dataset: lidar point, point from different image or a GIS point. Its transform to an image is \( x' = x'(p) \), where the elements of \( p \) are the exterior orientation parameters, relative or absolute. Given an edge map \( \mathcal{I} \) of the image, let \( P(x'(p)|\mathcal{I}) \) to be the probability that the point \( x \) will be transformed to the correct edge \( \mathcal{I} \). In other words, it is the probability that the point \( x \) is on the edge \( \mathcal{I} \) and called the likelihood of \( p \). The power of this model comes from the fact that the distribution of the error model could be any distribution, thus a stochastic model can be built instead of a deterministic model. For simplicity, a normal Gaussian distribution is used based on the findings of [137].

Let there be two sets of edge points \( x_i \in \mathbb{R}^2 \) and \( X_i \in \mathbb{R}^3 \) which respectively belong to the image data and ground data. The 3D edge points \( X_i \) are assumed absolute and are adjusted, and their projections to the image will only conform to the image edges upon successful adjustment of the camera parameters. The projections from ground
data to image domain is performed by applying the collinearity equations [182]:

\[
x'_i = x_0 - f \left( \frac{(X_i - X_c)r_{11} + (Y_i - Y_c)r_{12} + (Z_i - Z_c)r_{13}}{(X_i - X_c)r_{31} + (Y_i - Y_c)r_{32} + (Z_i - Z_c)r_{33}} \right),
\]

\[
y'_i = y_0 - f \left( \frac{(X_i - X_c)r_{21} + (Y_i - Y_c)r_{22} + (Z_i - Z_c)r_{23}}{(X_i - X_c)r_{31} + (Y_i - Y_c)r_{32} + (Z_i - Z_c)r_{33}} \right),
\]

where \(X_i = (X_i, Y_i, Z_i)\) are ground points, \((X_c, Y_c, Z_c)\) is the perspective center, \((x_0, y_0)\) is the principal point coordinates (PPO), \(f\) is focal length, and \(r_{ij}\) is the component at the \(i^{th}\) row and \(j^{th}\) column of the rotation matrix taking orientation angles \(\omega, \phi, \kappa\) as its parameters. The equations given in (4.11) and (4.12) are based on the principle that the perspective center, image point and corresponding 3D point are collinear and provides a relation between the camera parameters, image and the 3D points. Hence, the parameters of the model include \(p = [\omega, \phi, \kappa, X_c, Y_c, Z_c, f, x_0, y_0]\).

Note that as stated, the image distortion and refraction in this model are removed, hence they are not part of the collinearity equation.

In traditional photogrammetric solution, the 3D points are the GCPs, and their conjugate image point coordinates are known and are used to adjust the parameters vector \(p\). For this research, the ground data corresponding to lidar or GIS is treated as a set of 3D points which serve a similar purpose as GCPs. These points, however, do not have conjugate image points and establishing such conjugacies is nontrivial, but they are still projected to the image space, and the distance from the closest image points lying on extracted image edges is computed to these 3D projected points. The adjustment of the parameter vector \(p\) provides a means to minimize the distance between the projected and image points. In the following discussion is presented an intuitive, yet powerful technique to facilitate fast computation of the distances
between the projected and image edge points without performing a brute-force search used for dense matching.

Direct conjugacies between the image data and the ground data are not known, hence the distances computed between the projections of the ground data and the image data are treated as hypotheses. Note that the hypothesized distance for a projected ground datum also implies hypothesized corresponding image edge pixel. Since the initial camera parameters are approximate solutions, such implied correspondences, \(<x_j, x'_i>\), become random variables whose likelihood can be computed assuming a zero mean normal distribution \(N(0, \sigma)\):

\[
p(<x_j, x'_i>) \propto \frac{1}{\sqrt{2\pi}\sigma} \exp \left(-\frac{d(x_j, x'_i)^2}{2\sigma^2}\right),
\]

where \(d(.)\) is as given in (4.48) and \(\sigma\) is standard deviation of the distribution, which represents the uncertainty of observations and is set to a constant during the adjustment process. In the ideal case, when the ground and image data are precisely registered, the distances are 0 and the likelihood values are at their highest values, which is \(p(<x_j, x'_i>) = 1\). The choice of normal distribution in this model is a natural one assuming that the noise is normally distributed, such that the computed distances are concentrated around 0 in the case of precise registration.

The likelihood defined in (4.13) takes \(x'_i\) as its parameter which is computed using the collinearity equations (4.11) and (4.12). Considering that the camera parameters in the collinearity equations contain \(p = [\omega, \phi, \kappa, X_c, Y_c, Z_c, f, x_0, y_0]\), the likelihood
also takes $p$ as its parameter and can be written as:

$$p(<x_j, x'_i>) = p(x'(p)|D) = \frac{1}{\sqrt{2\pi}\sigma} \exp\frac{-D(x'_i(p))^2}{2\sigma^2}. \quad (4.14)$$

Applying Bayes’ rule to the conditional probability stated in this equation we have:

$$p(x'(p)|D) = \frac{p(D)}{p(x'(p))}p(D|x'(p)). \quad (4.15)$$

In (4.15), the probability of generating a distance transform has a uniform distribution and the probability of projection of any ground data is equally likely, hence is constant, $C$, such that $p(x'(p)|D) = Cp(D|x'(p))$. The posterior probability of distance transform $D$ given projected ground data is computed based on the image data using (4.13) and (4.14) and is computed from:

$$p(x'(p)|D) = \frac{C}{\sqrt{2\pi}\sigma} \exp\frac{-1}{2\sigma^2} D(x'(p))^2. \quad (4.16)$$

Note that the likelihood given in (4.16) is not limited to normal distribution and can be replaced with other continuous distributions to model systematic camera errors or observation errors with known behavior.

The projection of edges from the lidar data or the sampled GIS data will result in multiple points $\tilde{x} = \{x'_1, x'_2, \ldots, x'_N\}$ where $N$ is the number of projected ground points. Under the assumption that the ground data is observed independently, the probability given in (4.16) can be extended to all points by:

$$p(\tilde{x}(p)|D) = p(x'_1(p), x'_2(p), \ldots, x'_N(p)|D) = \prod_{i=1}^{N} p(x'_i(p)|D). \quad (4.17)$$
Substituting (4.16) into (4.17), the likelihood of observing a set of projections for a given distance transform becomes:

\[ p(\tilde{x}(p)|D) = \prod_{i=1}^{N} \frac{C}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}D(x'(p))^2\right) \]  

(4.18)

\[ = \tilde{C} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{N} D\left(x'(p) \right)^2\right) \]  

(4.19)

where \( \tilde{C} = \left(\frac{C}{\sqrt{2\pi}\sigma}\right)^N \) is a constant. In this paper, our goal is to estimate the camera parameter vector \( p \), such that, when \( p \) is precise, the likelihood in (4.19) is maximum. Hence, we define our cost function using the likelihood term in (4.19). Considering that \( C \) and \( \tilde{C} \), are constants, maximizing this likelihood term is equivalent to maximizing the posterior probability given in the right side of the equal sign in (4.19).

Equivalently, the maximum a posteriori estimate defined above can be converted to a minimization problem by taking the negative logarithm of both parts of (4.19). A benefit of this minimization problem is its numerical stability due to summation instead of multiplication of small probability values:

\[ l(p) = -\log p(\tilde{x}(p)|D) = -\log \tilde{C} - \left(\frac{-1}{2\sigma^2} \sum_{i=1}^{N} D\left(x'(p) \right)^2\right) \]  

(4.20)

where \( l(.) \) is the posterior term. Using this term the solution to the camera parameter is given as:

\[ \hat{p} = \arg\max_{p} p(\tilde{x}(p)|D) = \arg\min_{p} l(a) \]  

(4.21)

\[ = \arg\min_{p} -\log \tilde{C} + \frac{1}{2\sigma^2} \sum_{i=1}^{N} D\left(x'(p) \right)^2 \]  

(4.22)

\[ = \arg\min_{p} \sum_{i=1}^{N} D\left(x'(p) \right)^2. \]  

(4.23)
In this formulation, the estimate \( \hat{p} \) can be computed following traditional minimization by setting the first derivative of the function to 0.

In addition to determining the most likely camera parameters, the uncertainty in the localization in terms of both the variance of the estimated positions and the probability that a qualitative failure has occurred can be estimated. The uncertainty is estimated by measuring the rate at which the likelihood function falls off from the peak. This is performed by fitting a normal distribution to a neighborhood of values in the likelihood function around the location of the maximum. The probability of failure is estimated by comparing the sum of the likelihood scores under the selected peak with the remainder of the parameter space [122].

**Frequentist (least squares, LSE) model**

This maximum a posteriori estimate problem resembles that of the least squares estimate problem in the case when there no outliers (non existent ground data), a constant variance (\( \sigma \)) and independent observations with normally distributed error [117, 122] and [86]. The scheme follows the same rules defined above if another model is selected. Rewriting (4.23) in terms of the minimization problem results in:

\[
\epsilon = \sum_{i=1}^{N} D(x'(p))^2,
\]

and corresponds to the model given by:

\[
0_{N \times 1} = d(p) + e,
\]
where \( \mathbf{e} \) is the error vector and \( \mathbf{d} \) vector is given by: 
\[
\mathbf{d} = \{ d_i \mid d_i = \mathcal{D}(x'_i(p)) \}, \quad 1 \leq i \leq N
\]

The solution to (4.25) can be computed by first linearizing (4.25) and then estimating the parameters iteratively starting from an initial estimate, \( p_0 \):

\[
0 = \mathbf{d}(p_0) + J(d(p)) \bigg|_{p=p_0} \Delta p + \mathbf{e},
\]

where \( J(d(p)) \) is the Jacobian of the vector-valued function \( \mathbf{d} \) and is a \( N \times 9 \) matrix in our problem. The projected ground data using the collinearity equations, \( x'_i(p) = (x'_i, y'_i) \) when introduced to \( J \), each row will have the following construct:

\[
J(d_i) = \begin{bmatrix}
\frac{\partial}{\partial x} \mathcal{D} \bigg|_{x=x'_i} & \frac{\partial}{\partial y} \mathcal{D} \bigg|_{x=x'_i}
\end{bmatrix}_{1 \times 2} \begin{bmatrix}
\frac{\partial x'_i}{\partial p} \\
\frac{\partial y'_i}{\partial p}
\end{bmatrix}_{2 \times 9}.
\]

Therefore, the solution of \( \Delta p \) is estimated using the generic least squares estimate as:

\[
\Delta p = -(J^T J)^{-1} J^T \mathbf{d}(p_0).
\]

For a more stable solution with better convergence, the generic least squares solution can be altered with the Levenberg-Marquardt dampener:

\[
\Delta p = -(J^T P + \lambda \text{diag}(J^T P J))^{-1} J^T P \mathbf{f}(p_0),
\]

where \( P \) is the weight matrix with entries computed from the distance, \( \lambda \) is the damping factor value used during minimization, \( \text{diag}(.) \) refers to the diagonal of the corresponding matrix. A large \( \lambda \) value (\( \sim 10 \)) provides a stable, but slow convergence; while a lower \( \lambda \) allows for faster convergence, but may be unstable depending on the initial parameter values. When \( \lambda = 0 \) the solution reduces to least squares.
solution. The solution provided in (4.28) allows the use of any type of data, not just image data, but the discrete points of the lidar data, or the points extracted from the polyline/polygon sets of GIS data. There is no need to find smooth patches or normals or parametric models for linear data.

**LSE linearization of collinearity equation[118]**

The collinearity equation will be used to project the ground points to the image. To solve for the camera orientation parameters with redundant data, this is a non-linear equation, and the collinearity equations need to be linearized. The initial approximations for the solution will be the initial GPS/IMU data.

The collinearity equation links the conjugate ground points to the image points. For nine unknowns and two collinearity equations, one for $x$ and one for $y$, the redundancy budget for a ground control based solution is $r = 2 \times n - 9$, where $n$ = number of points. This becomes irrelevant using lidar data, as thousands of lidar points are used.

Since the collinearity equations are non-linear functions with respect to most of the variables, it is necessary to linearize them so that they can be simultaneously solved for the unknowns; in addition, in the linear form, they can be developed into a set of normal equations enabling a least squares solution. The Taylor power series expansion can be used to expand a continuous monotonic function of the general form of equation (4.30):

\[
    f(x) = \sum_{k=0}^{\infty} \frac{f^k(a)}{k!} (x - a)^k,
    \tag{4.30}
\]

where $a$ is a known point around which the series is centered and $f^k$ denotes the $k^{th}$ derivative of $f(x)$, evaluated at $a$. Theoretically, this would be summed out to infinity,
however, for values of $x$ close to $a$, all but the first few terms become insignificant. Thus a good approximation is using the first two terms \[182\], equation\((4.31)\):

$$f(x) \sim f(c) + f'(c)(x - c). \quad (4.31)$$

For a function of more than one variable, the same approximation applies, with the first derivative term being replaced by terms representing the partial derivative of $f(x)$ with respect to each variable, equation (4.32):

$$f(a, b, c...) \sim f(a_0, b_0, c_0) + \frac{\partial f}{\partial a}(x - a) + \frac{\partial f}{\partial b}(x - b) + \frac{\partial f}{\partial c}(x - c) + ..., \quad (4.32)$$

where $a_0$, etc. represents the initial estimated values where the partial derivatives are evaluated. The two collinearity equations, equation (2.14) and equation (2.15), can be similarly expanded. The linearization terms for the nine unknowns of the interior and exterior orientation, equation\((4.33)\) and equation\((4.33)\):

$$x = x_0 + \frac{\partial x}{\partial \omega_0} (\omega - \omega_0) + \frac{\partial x}{\partial \phi_0} (\phi - \phi_0)$$

$$+ \frac{\partial x}{\partial \kappa_0} (\kappa - \kappa_0) + \frac{\partial x}{\partial X_{c0}} (X_c - X_{c0})$$

$$+ \frac{\partial x}{\partial Y_{c0}} (Y_c - Y_{c0}) + \frac{\partial x}{\partial Z_{c0}} (Z_c - Z_{c0})$$

$$+ \frac{\partial x}{\partial x_{00}} (x_0 - x_{00}) + \frac{\partial x}{\partial y_{00}} (y_0 - y_{00}) + \frac{\partial x}{\partial f_0} (f - f_0),$$
\[ y = y_0 + \frac{\partial y}{\partial \omega_0} (\omega - \omega_0) + \frac{\partial y}{\partial \phi_0} (\phi - \phi_0) \\
+ \frac{\partial y}{\partial \kappa_0} (\kappa - \kappa_0) + \frac{\partial y}{\partial X_c} (X_c - X_{c0}) \\
+ \frac{\partial y}{\partial Y_c} (Y_c - Y_{c0}) + \frac{\partial y}{\partial Z_c} (Z_c - Z_{c0}) \\
+ \frac{\partial y}{\partial x_0} (x_0 - x_{00}) + \frac{\partial y}{\partial y_0} (y_0 - y_{00}) + \frac{\partial y}{\partial f_0} (f - f_0). \]

Here each of the partial derivatives is evaluated using the initial values of the variables and each of the terms of the form \((\text{var} - \text{var}_0)\) can be considered the corrections to be applied to the initial values, and are the unknowns that are solved for. \(x_0\) and \(y_0\) represent the full respective functions evaluated using all of the initial values, both fixed and estimated. Note that only unknown variables of the functions appear in the partial differential terms, since any known values are held fixed, and so their correction factors become zero. These linear equations can more simply be expressed, including a variance factor to allow for a least squares solution, as equation (4.33) and equation (4.34):

\[ x - x_0 + v_x = d_{11} \delta \omega + d_{12} \delta \phi + d_{13} \delta \kappa + d_{14} \delta X_c + d_{15} \delta Y_c + d_{16} \delta Z_c + d_{17} \delta x_0 + d_{18} \delta y_0 + d_{19} \delta f, \] (4.33)

\[ y - y_0 + v_y = d_{21} \delta \omega + d_{22} \delta \phi + d_{23} \delta \kappa + d_{24} \delta X_c + d_{25} \delta Y_c + d_{26} \delta Z_c + d_{27} \delta x_0 + d_{28} \delta y_0 + d_{29} \delta f. \] (4.34)

The linearization can be programmed quite easily and optimized for processing by the following [25]:

For simplicity, the collinearity equations, eq. (2.14) and eq. (2.15) will be shown as:

\[ x = x_0 - \frac{U}{W}, \] (4.35)
\[ y = y_0 - \frac{V}{W}. \]  \hfill (4.36)

The design matrix appears as:

\[
B = \begin{pmatrix}
\frac{\partial x}{\partial x_0} & \frac{\partial x}{\partial y_0} & \frac{\partial x}{\partial c} & \frac{\partial x}{\partial X_c} & \frac{\partial x}{\partial Y_c} & \frac{\partial x}{\partial Z_c} & \frac{\partial x}{\partial \omega} & \frac{\partial x}{\partial \phi} & \frac{\partial x}{\partial \kappa} \\
\frac{\partial y}{\partial x_0} & \frac{\partial y}{\partial y_0} & \frac{\partial y}{\partial c} & \frac{\partial y}{\partial X_c} & \frac{\partial y}{\partial Y_c} & \frac{\partial y}{\partial Z_c} & \frac{\partial y}{\partial \omega} & \frac{\partial y}{\partial \phi} & \frac{\partial y}{\partial \kappa} \\
\end{pmatrix}
\]  \hfill (4.37)

The first row represents the partial derivatives of eq. (2.14) and the second row represents the partial derivatives of eq. (2.15). The first three columns represent the partial derivatives with respect to the interior orientation, \( x_0, y_0, c \), the principal point coordinates and the focal length. The middle six columns represent the partial derivatives with respect to the exterior orientation parameters: the perspective center coordinates \( X_c, Y_c, Z_c \) and the orientation angles \( \omega, \phi, \kappa \). The last three columns represent the partial derivatives with respect to the survey points in question, in this case the lidar data points.

Partial derivatives with respect to interior orientation, \( x_0, y_0, c \) only:

\[
\frac{\partial x}{\partial x_0} = 1, \quad \frac{\partial x}{\partial y_0} = 0, \quad \frac{\partial x}{\partial c} = -\frac{U}{W},
\]

\[
\frac{\partial y}{\partial x_0} = 0, \quad \frac{\partial y}{\partial y_0} = 1, \quad \frac{\partial y}{\partial c} = -\frac{V}{W}.
\]  \hfill (4.38)

Partial derivatives taken with respect to perspective centers, where \( P \) represents the parameter in question:

\[
\frac{\partial x}{\partial P} = -c \frac{W \frac{\partial U}{\partial P} - U \frac{\partial W}{\partial P}}{W^2} = -\frac{c}{W} \left( \frac{\partial U}{\partial P} - \frac{U \partial W}{W \partial P} \right),
\]  \hfill (4.39)

\[
\frac{\partial y}{\partial P} = -c \frac{W \frac{\partial V}{\partial P} - V \frac{\partial W}{\partial P}}{W^2} = -\frac{c}{W} \left( \frac{\partial V}{\partial P} - \frac{V \partial W}{W \partial P} \right).\]  \hfill (4.40)
For the perspective center coordinates, the partial derivatives of functions $U, V, W$:  

\[
\begin{align*}
\frac{\partial U}{\partial X_c} &= -m_{11} & \frac{\partial U}{\partial Y_c} &= -m_{12} & \frac{\partial U}{\partial Z_c} &= -m_{13}, \\
\frac{\partial V}{\partial X_c} &= -m_{21} & \frac{\partial V}{\partial Y_c} &= -m_{22} & \frac{\partial V}{\partial Z_c} &= -m_{23}, \\
\frac{\partial W}{\partial X_c} &= -m_{31} & \frac{\partial W}{\partial Y_c} &= -m_{32} & \frac{\partial W}{\partial Z_c} &= -m_{33}.
\end{align*}
\]

(4.41)

Thus the partial derivatives of the functions $x$ and $y$ are, with respect to perspective center:

\[
\begin{align*}
\frac{\partial x}{\partial X_c} &= -cW (-m_{11} + \frac{U}{W} m_{31}) & \frac{\partial y}{\partial X_c} &= -cW (-m_{21} + \frac{U}{W} m_{31}), \\
\frac{\partial x}{\partial Y_c} &= -cW (-m_{12} + \frac{U}{W} m_{32}) & \frac{\partial y}{\partial Y_c} &= -cW (-m_{22} + \frac{U}{W} m_{32}), \\
\frac{\partial x}{\partial Z_c} &= -cW (-m_{13} + \frac{U}{W} m_{33}) & \frac{\partial y}{\partial Z_c} &= -cW (-m_{23} + \frac{U}{W} m_{33}).
\end{align*}
\]

(4.42)

The partial derivatives of the orientation matrix with respect to the angles, where $M$ is the same as in eq.(2.14):

\[
\frac{\partial M}{\partial \omega} = M_\kappa M_\phi \frac{\partial M_\omega}{\partial \omega} = M \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix},
\]

(4.43)

\[
\frac{\partial M}{\partial \phi} = M_\kappa \frac{\partial M_\phi}{\partial \phi} M_\omega = M \begin{pmatrix} 0 & \sin \omega & -\cos \omega \\ -\sin \omega & 0 & 0 \\ \cos \omega & 0 & 0 \end{pmatrix},
\]

(4.44)

\[
\frac{\partial M}{\partial \kappa} = \frac{\partial M_\kappa}{\partial \kappa} M_\phi M_\omega = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} M.
\]

(4.45)
Thus the partial derivatives with respect to the orientation angles:

\[
\begin{pmatrix}
\frac{\partial U}{\partial P} \\
\frac{\partial V}{\partial P} \\
\frac{\partial W}{\partial P}
\end{pmatrix} = \frac{\partial M}{\partial P} \begin{pmatrix}
\frac{\partial U}{\partial P} \\
\frac{\partial V}{\partial P} \\
\frac{\partial W}{\partial P}
\end{pmatrix},
\]

where \( P \) represents \( \omega, \phi, \kappa \) orientation angles.

Evaluation of the partial derivatives of \( x \) and \( y \) with respect to the orientation angles:

\[
\begin{align*}
\frac{\partial x}{\partial \text{angle}} &= -c_W \left( \frac{\partial U}{\partial \text{angle}} - U \frac{\partial W}{\partial \text{angle}} \right), \\
\frac{\partial y}{\partial \text{angle}} &= -c_W \left( \frac{\partial V}{\partial \text{angle}} - V \frac{\partial W}{\partial \text{angle}} \right).
\end{align*}
\]

Finally, the partial derivatives with respect to the surveyed points are the negative of the partial derivatives with respect to the perspective centers.

**LSE linearization of distance function**

The distance function is a function of the collinearity equation, in that the laser points are projected onto the image using the collinearity equation. Thus, minimizing the distance between the edge pixels to the nearest laser point is a function of the collinearity equation. The distance function is the standard Euclidean function. For simplicity, the square of the distance function is used, leaving out the square-root:

\[
d = (x - x_e)^2 + (y - y_e)^2,
\]

where \((x_e, y_e)\) is the coordinate pair of the closest edge pixel to the projected laser point coordinate pair \((x, y)\), which are calculated using the collinearity equation. The
linearization of the distance function with respect to interior orientation, perspective center, orientation angles and surveyed points is necessary. As section 4.3.1 contained the linearization method, all that is necessary is to use the chain-rule to obtain the partial derivatives for the distance function.

4.3.2 Distance function advantages

Using this solution, any type of data can be used, not just image data, but the discrete points of the lidar data, or the points extracted from the polyline/polygon sets of GIS data. There is no need to find smooth patches or normals or parametric models for linear data.

![Image 425 (3-D Probability function of the image edge map, based on (4.13))](image)

Figure 4.4: The probability surface of a building from image 425 Hough edge map; the Hough transformed image is shown on fig. (6.2). Each edge is fitted or superimposed with a normal distribution, to represent the uncertainty in the edges. Red color represents probabilities closer to edges, blue represents probabilities less likely to be edges; the vertical coordinate represents the probability, with the maximum of 1.
The maximum-likelihood measure gains robustness by explicitly modeling the possibility of outliers and allowing matches against pixels that do not precisely overlap the template pixel [122]. The least squares solution provides a standard photogrammetric solution for non-overlapping pixels.
Chapter 5

FEASIBILITY STUDIES

Photogrammetry is the traditional method of surface reconstruction [9]. The surface is reconstructed from two overlapping aerial images — a process known as stereopsis. This requires the identification of the ground features/points in both images, as well as the exterior orientation of the images: the registration problem. These steps are necessary to orient the images into the correct positions they were at the time of photography [147]. The crucial step in stereopsis is the identification of the same ground feature/point, also referred to as a correspondence problem or image matching. Human operators are remarkably adept in finding conjugate, or identical, features. The surfaces generated by operators on analytical plotters or on softcopy workstations are of high quality but the process is time and cost intensive. Thus, major research efforts have been devoted to make stereopsis an automatic process. The success of automatic surface reconstruction from aerial imagery is marginal. Despite considerable research efforts there is no established and widely accepted method that would generate surfaces in more complex settings, say large-scale urban scenes, completely, accurately, and robustly. A human operator needs to be involved, at least on the level of quality control and editing [9]. In contrast to this, laser data inherently gives horizontal/vertical coordinates, however, there is no semantic information
available from just a set of points, and as the pulse rate of laser range sensors continues to increase rapidly, it produces huge datasets. Despite this, laser range data is now being routinely fused with data from other sensors, especially multispectral and hyperspectral cameras [101]. While laser range data is not a product of any photogrammetric application, it is a very useful product when combined/fused with image data, and to a lesser degree this is true with radar data, but radar is, generally, less accurate [76]. Thus, the combination of range data and aerial data compensates each other in their disadvantages, and this input is what the proposed method will use for registering the data to each other.

5.1 Data description

In order to show the performance of the proposed approach, two experimental sites are selected: one in California, San Bernardino (see Figure 5.9) and one at the Ohio State University Campus (see Figure 5.7 and 5.8). The GPS/IMU solutions for the camera orientations, as well as the initial camera calibration parameters, are provided with the images. These parameters are used as approximate initial solutions and are shown to improve when fused with the ground data in the form of lidar point cloud or discretized GIS vector data. The accompanying lidar and GIS data for the California site is provided from the San Andreas fault mapping project [171] and from San Bernardino county data and personal measurements on site, while the Ohio State site data are obtained from the Ohio Geographically Referenced Information Program (OGRIP) [120] and the Franklin County Auditor.

The proposed method has been tested with a set of multispectral images and lidar data. The multispectral images were taken with a Redlake MS 4100 Ducantech
high resolution 3-CCD Digital Multispectral Camera in May 2005 in San Bernardino, California, near the San Bernardino Fwy., I-10, at the intersection of Commercenter St. and Hospitality Ln, as shown in Figure (5.1) from Google Maps. The camera can be configured as a regular RGB (color) camera or a Color Infrared Imaging (CIR) camera. The multispectral image used in this experiment was taken under the CIR configuration. Under the configuration, it can have Red, Green, and Near Infrared bands from 400-1000 nm. The size of an image is $1920 \times 1080$ pixels at 14.2 mm $\times$ 8 mm and the focal length of the camera is 14mm [132]. The laser range data were acquired with an Optech ALTM3100 with a built-in GPS/IMU Positioning unit also
Table 5.1: ALTM3100 parameters [75]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal Accuracy</td>
<td>$1/5, 500 \times \text{altitude (m AGL)}; 1 \sigma$</td>
</tr>
<tr>
<td>Elevation Accuracy</td>
<td>&lt;5-20 cm; 1 $\sigma$</td>
</tr>
<tr>
<td>Effective Laser Repetition Rate</td>
<td>Programmable, 33..100 kHz</td>
</tr>
<tr>
<td>Range Capture</td>
<td>Up to 4 range measurements, including 1st, 2nd, 3rd, last returns</td>
</tr>
<tr>
<td>Intensity Capture</td>
<td>12-bit dynamic range measurements for all recorded returns, including last return</td>
</tr>
<tr>
<td>Scan FOV</td>
<td>$0 – 50^\circ$ (Programmable in $\pm 1^\circ$ increments.)</td>
</tr>
<tr>
<td>Scan Frequency</td>
<td>30 to 70 Hz ($&gt;70$ Hz optional) programmable in 1 Hz increments</td>
</tr>
<tr>
<td>Roll Compensation</td>
<td>$\pm 5^\circ$; more compensation available if FOV reduced, programmable in $\pm 1^\circ$ increments</td>
</tr>
<tr>
<td>Position and Orientation System</td>
<td>POS AV 510 OEM includes embedded BD960 GNSS receiver (GPS and GLONASS)</td>
</tr>
<tr>
<td>Beam Divergence</td>
<td>Dual divergence: 0.25 mrad (1/e) and 0.8 mrad (1/e) , nominal</td>
</tr>
</tbody>
</table>

on May 2005 of the same area, about 1 m ground sample distance. The raw lidar data were processed to generate data points in the UTM 11N (Universal Transverse Mercator Zone 11 North) projection system based on the WGS-84 (World Geodetic Survey) ellipsoid and datum. The unit of measurement is thus meters. Each record of the laser range data consists of $x, y, z$ coordinates and intensity values. The other site was The Ohio State University site in Columbus Ohio, the images were taken with a Jena LMK 2015 camera, with onboard GPS/IMU. The size of an image is $9984 \times 9728$ pixels or $9 \times 9$ inches and the focal length of the camera is 152.152 mm [174], with principal point coordinates $x_0=-0.015$ mm and $y_0=0.012$ mm. The film is configured for black and white imaging. The laser range data were acquired from the State of Ohio, Geographically Referenced Information Program [120], about
2 m ground sample distance. The raw lidar data were processed to generate data points in Ohio State Plane South projection system based on the NAD-83 (North American Datum) datum. The unit of measurement is feet. Each record of the laser range data consists of $x, y, z$ coordinates and intensity values.

The laser range data covered by the California multi-spectral images are shown in Figures (5.2, 5.3 & 5.4 and 5.5). The camera was oriented using a GPS/IMU positioning unit. The laser system assumes the use of a high-quality direct sensor orientation system, which provides the exterior orientation data.

Surveying of densely built-up urban areas is in high demand and yet this is one of the most difficult mapping tasks to perform. This is primarily due to the large number of manufactured objects with lots of vertical surfaces, occlusions, shadows and moving objects. Probably the surface discontinuities, generally called break lines, represent the most difficult problem, and from a strictly theoretical point of view, they would require a diminishingly small sampling distance. Consequently, this is the case where anything that can increase the sampling frequency for the laser system is appreciated. Multiple laser returns, which are used primarily for vegetation separation, can virtually increase sampling rate locally by providing two or more observations for one laser pulse; for example, from the ground and from rooftops [170]. However, this is a very rare scenario, since the probability that the laser beam hits the edge of a building is very small. Therefore, the only way to introduce additional information into the process is the use of simultaneously acquired imagery, and processing the imagery for features like edges.
The edge detection from the images was acquired using the Edge Detection and Image SegmentatiON (EDISON) System ver1.0 \(^3\), a mean shift segmentation method [32], and the Canny edge detector. EDISON was used for the initial, coarse adjustment of the camera parameters. For a refined adjustment, the Canny edge detector was used, as this is more sensitive to individual lines, like road markings or curbs and is a leading detector [157]. The curb data can then be matched to the GIS data. EDISON has the advantage in detecting large, closed contours, and is not as sensitive to individual lines, making it ideal for building detection. Edge detection is a non-trivial task: to state a certain threshold on the magnitude of the intensity change between two neighboring pixels to assume that there is an edge between these pixels is not simple [98]. Rockett [137] concluded that the localization error of the Canny edge detector is a Gaussian function. Note that both coarse and fine level edges extracted from the image data is passed through the distance transform following the discussion in Section 3.1.1.

EDISON was developed by the Robust Image Understanding Laboratory at Rutgers University. It is a low-level vision tool that performs edge detection and image segmentation. It is packaged under a platform independent graphical user interface. The EDISON system was used, since it creates closed edges from the segmentation using the mean-shift method, a necessity in the case of buildings, where the outline of the building is always closed, except if it falls on the edge of the image, and is partially cut. The edge detection is then a result of the image segmentation, where a closed contour is drawn around the segmented areas. Based on the focal length and flying height above ground of 650 m, and the pixel size, an 85 pixel length edge is

\(^3\)http://www.caip.rutgers.edu/riul/research/code/EDISON/index.html
about 30 m on the ground. Any object with that size is a good chance of being a building or sidewalk curb edge. Any longer edge would mean a very big building or long sidewalk edge, so 30 m is a good choice for a generic office building size. Adjustment is done by selecting areas that are at least 30 m $\times$ 30 m, or about 7500 pixels. Only areas with 7500 or more pixels were considered, to remove cars, smaller trees and road markings. The mean shift algorithm seeks the “mode” or point of highest density of a data distribution.

Mean Shift Algorithm: [187]

1. Choose a search window size.

2. Choose the initial location of the search window.

3. Compute the mean location (centroid of the data) in the search window.

4. Center the search window at the mean location computed in Step 3.

5. Repeat Steps 3 and 4 until convergence.

Mean Shift Segmentation Algorithm:

1. Convert the image into tokens (via color, gradients, texture measures, etc.).

2. Choose initial search window locations uniformly in the data.

3. Compute the mean shift window location for each initial position.

4. Merge windows that end up on the same “peak” or mode.

The Canny edge detector is a multi-step edge detection procedure by Canny [26]. Canny was used, since it is able to detect edges with noise suppressed at the same time, and it is a well known operator [179]:

\[ 129 \]
1. Smooth the image with a Gaussian filter, $G$, to reduce noise and unwanted details and textures.

$$g(m, n) = G_\sigma(m, n) * f(m, n)$$

where

$f(m, n)$ is the image pixel $(m, n)$

and

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{m^2 + n^2}{2\sigma^2}}$$

2. Compute gradient of $g(m, n)$ using any of the gradient operators: Roberts, Sobel, Prewitt.

$$M(m, n) = \sqrt{g_m^2(m, n) + g_n^2(m, n)}$$

and

$$\theta(m, n) = \tan^{-1}[g_n(m, n)/g_m(m, n)]$$

3. Threshold $M$:

$$M_T(m, n) = \begin{cases} 
M(m, n) & \text{if } M(m,n)>T \\
0 & \text{otherwise}
\end{cases}$$

where $T$ is so chosen that all edge elements are kept, while most of the noise is suppressed.

4. Suppress non-maxima pixels in the edges in $M_T$ obtained above to thin the edge ridges, as the edges might have been broadened in step 1. To do so, check to see whether each non-zero $M_T(m, n)$ is greater than its two neighbors along the gradient direction $\theta(m, n)$. If so, keep $M_T(m, n)$ unchanged, otherwise, set it to 0.
5. Threshold the previous result by two different thresholds $\tau_1$ and $\tau_2$ (where $\tau_1 < \tau_2$) to obtain two binary images $T_1$ and $T_2$. Note that compared to $T_1$, $T_2$ has less noise and fewer false edges, but larger gaps between edge segments.

6. Link edges segments in $T_2$ to form continuous edges. To do so, trace each segment in $T_2$ to its end and then search its neighbors in $T_1$ to find any edge segment in $T_1$ to bridge the gap until reaching another edge segment in $T_2$.

One problem with the basic Canny operator is to do with Y-junctions: places where three ridges meet in the gradient magnitude image. Such junctions can occur where an edge is partially occluded by another object. The tracker will treat two of the ridges as a single line segment, and the third one as a line that approaches, but doesn’t quite connect to that line segment.

To register images, two data modalities are used, including lidar and GIS, as shown in Figure 5.9 for the California study site.

Similar to the segment boundaries of lidar elevation data, GIS contains vector data denoting the edges of surface features manually extracted by an operator. GIS data is usually orthographic in the form of a map and does not contain side views of buildings, hills or other tall objects. As a common practice, GIS data generally contains no elevation information; hence, the data cannot be directly projected to the image space. There are two possible ways to offset this problem so that the GIS data can be registered with the image. One possibility is to use the lidar data to define the GIS elevation. A good GIS data should register very closely with the lidar data, as illustrated in Figure 5.10a and b, so that complementary information can be easily and correctly transferred.
5.1.1 Camera orientation parameters from GIS data combined with lidar elevation data

For indirect orientation, it is possible to use information extracted from other datasets such as LIDAR and GIS. While very accurate GIS and LIDAR datasets are publicly available from USGS (United States Geological Survey) and state agencies for any given area, they are usually not used for registration due to the fact that establishing correspondences between an image and GIS/LIDAR is a challenging problem and is an ongoing research [69] [156]. In recent years, researchers have exploited the features extracted from the LIDAR data, such as lines and planes, to register them with the images [63] [61] [80] [9]. Extracting linear and planar features from lidar data, however, is not always intuitive due to surface patterns, noise and LIDAR point density. In contrast to registration of LIDAR with images, the registration of GIS data with images has received less attention except only a few attempts such as [153, 155] and [29]. GIS features and the features extracted from a different sensory data may not be the same due to different sampling of the real world [148]. The number of vertices that constitute a shape can also make features to look different. This research demonstrates a novel method to register an image with the GIS and LIDAR data without establishing explicit correspondences by eliminating the feature matching step.

Similar to the segment boundaries of LIDAR elevation data, GIS contains vector data denoting the edges of surface features manually extracted by an operator. GIS data is usually orthographic in the form of a map and does not contain side views of buildings, hills or other tall objects. As a common practice, GIS data generally contains no elevation information; hence, cannot be directly projected to the image
space. There are two possible ways to offset this problem so that the GIS can be registered with the image. One possibility is to use the LIDAR data to define the GIS elevation. The registration of the GIS and lidar data buildings on the Ohio State campus or one of the buildings in the examined dataset, fits almost perfectly to each other figure(5.10). This is an ideal case, since there are no trees or other objects that would interfere or create noise with the lidar returns. The building outlines are crisp and clear. This allows the extraction of elevation data to the GIS line data. In case there is noise, the intensity values of the lidar returns, if available, may assist in delineating the features. However, even this may not always assist in delineating the features, as intensity values of different surface types may be indistinguishable.

In ArcGIS 9.3, in ArcToolbox under “Samples”>“Data Management”>“Features”, there is a tool called “Write Features to Text File” that will export selected feature vertices to a text file. Interpolating between the vertices in a linear fashion will densify the available ground points used for control in matching. Adding elevation values to the interpolated GIS points is a crucial step. This will done from the available lidar data, since most GIS features generally do not contain elevation data, except for some ground points or hypsography from digital line graphs (DLGs). The closest lidar point from inside the building polygon to the interpolated GIS point will represent the elevation value of the interpolated GIS point, even if the elevation is incorrect. The incorrect elevation may be identified after adjustment, as this point will have a higher standard deviation value, and can be eliminated. Alternatively, the EOPs may be computed by generic GIS elevations of the area, since the images are close to being vertical images and thus ground points with the same \((x, y)\) coordinates will be projected to the same point on the image, with only minor offset.
This allows the ability to select the elevation from the closest lidar point residing inside the polygon/polyline data. Alternatively, a generic elevation measured from spot elevations on the ground can also be assigned to the GIS, such as from the elevation available from USGS maps. With available elevation data, the GIS data provides adequate information that can be fused with images for precise camera parameters estimation.

The edges in the image data, which include the edges from the depth discontinuities, are extracted using both the Canny and the mean-shift segmentation based edge detectors. These edge detectors provide results in different scales, which, as discussed in the next section, provides us with the ability to perform coarse-to-fine parameter estimate. As seen in Figure 5.11c, the mean shift based edge detector provides sparse but closed edge contours with less noise; hence it is suitable for estimating an initial approximation of the camera parameters. The drawback of the sparse edges, however, is its lack of detail. Once the initialization is performed, we use edges extracted from the Canny edge detector for fine estimation (see Figure 5.11b). For the edges detected by Canny, we perform edge linking to generate longer edge segments. It should be noted that the edges extracted from the image generally contain more edge segments than both the GIS and lidar edges. These additional edges may be observed due to texture, trees and vehicles and can be filtered based on their length. In this research, the length based filtering is performed by searching straight lines using the Hough transform [73]. In particular, the camera parameters are used to determine the threshold for removing short edge segments. For instance, for the California site, given a certain focal length, pixel size and flying height above ground of 650 m, an 80-pixel long edge corresponds to 30 m on the ground. Any object with that size
has a good chance of being a building or sidewalk curb edges. It is conjectured that longer edges correspond to buildings or long sidewalk edges. Hence, the threshold is 30 m for a generic building size. After edges are pruned the camera parameters are adjusted by selecting areas that are at least $30 \times 30$ m, or about 7500 pixels. An illustration of this process is shown in Figure 6.2b and c, where the threshold is set at 80 pixels. For the Ohio State study site, the images are captured at flying height over 2200 m and have a smaller scale. A 50 m threshold was used to remove outlier edges.

After the post processing, the resulting edge segments along with the edges extracted from the lidar or GIS modalities serve as the data to estimate precise camera parameters. Considering missing and incomplete edge segments across modalities, establishing correspondences is a very challenging task. In the following discussion, the proposed approach is introduced that eliminates the requirement to establish correspondences between these features, yet provides precise camera parameters.
Figure 5.2: (a) Image 425 and (b) Canny edge detection (c) Edge detection after segmentation by EDISON/mean shift
Figure 5.3: (a) Image 426 and (b) Canny edge detection (c) Edge detection after segmentation by EDISON/mean shift
Figure 5.4: (a) Image 427 and (b) Canny edge detection (c) Edge detection after segmentation by EDISON/mean shift
Figure 5.5: Raw lidar of California area, showing the Welcome Center. Correct $(x, y, z)$ positions
Figure 5.6: (a) California welcome center (CWC on street map) [58] (b) Location with respect to Google Maps
Figure 5.7: (a) Image 40-01 used in the experiments for the OSU study site. The “Horse Shoe” shaped Ohio stadium is on the middle left of the image. (b) Image 39-014 used in the experiments for the OSU study site. The “Horse Shoe” shaped Ohio stadium is on the top left of the image.
Figure 5.8: (a) OSU GIS data (cropped for better visualization) that corresponds to part of the image in part (a). The GIS data contains the buildings, street centerlines and curbs; note that the stadium on the middle left. (b) Local lidar elevation image for the same study site with stadium in middle left. Only local data is shown for better visualization.
Figure 5.9: (a) Elevation thresholded lidar data for San Bernardino (b) GIS of San Bernardino local area with buildings, streets centerlines, curbs, parcels.
Figure 5.10: Ideal case for GIS and lidar data registration with no noise. GIS and lidar data from different sources register to each other quite well. Usually, this is not the case, since trees and lamps interfere and make lidar returns difficult to distinguish. (a) Near the Ohio State campus, the rectangular features represent the lidar data from the Ohio Geographically Referenced Information Program (OGRIP). The building outlines are from the Franklin County Auditor’s Office GIS. (b) Welcome Center Building near San Bernardino freeway, visible on images 425-427.
Figure 5.11: (a) Cropped image from the California study site image 425. (b) Edges detected using the Canny edge detector. Notice that some edges are not complete. (c) Closed edges detected using mean-shift segmentation.
Chapter 6

IMPLEMENTATION AND RESULTS

6.1 Implementation

The algorithm of the application of the distance transform is described for EO and RO using lidar data, using the collinearity equation to project the lidar points to the images. In both cases, the distance minimization takes place in image space, not object space. The image will then be the domain for the distance transform [17].

In the description of the algorithm, the projected “ground data” shape $P$ is moved to be registered, positioned in best alignment with a “image model” shape $X$. The data and the model shape may be represented in any of the data forms, then discretized into a common format; usually ground data are discrete data already determined from GPS/laser/topographic surveys or back-projection of GIS points, and the image model is either a digital camera image or aerial film and the film is scanned and rasterized to form the digital image. In the case of GIS data, the data needs to be discretized to individual points using sampling. The ground points are then projected onto the digital image. The number of discrete points in the data shape will be denoted $N_p$. Let $N_x$ be the number of discrete points involved in the model.
shape. The number of discrete points in each dataset may be different. The distance metric $d$ between an individual discrete data point $\mathbf{p}$ and a discrete model point $X$, not necessarily conjugate point, will be denoted in eq. (6.1) and eq. (6.2):

$$d(\mathbf{p},X) = \min_{\mathbf{x} \in X} \| \mathbf{x} - \mathbf{p} \|,$$  \hspace{1cm} (6.1)

where

$$\| \mathbf{x} - \mathbf{p} \| = \sqrt{\sum (x_i - p_i)^2},$$  \hspace{1cm} (6.2)

A point in $X$ may have more than one closest point in $P$. However, this is not a problem, as the adjustment applies to every point in $P$, thus all points will be adjusted, not just individual points. The closest point in $X$ that yields the minimum distance is denoted $\mathbf{y}$, such that $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{p}, X)$, where $\mathbf{y} \in X$. This minimum distance is already known by embedding the points into the distance transform, and the coordinates of the closest point is also known by default. Let $Y$ denote the resulting set of closest points, not necessarily conjugate points, and let $C$ be the closest point operator in eq. (6.3):

$$Y = C(P,X)$$  \hspace{1cm} (6.3)

Given the resultant corresponding point set $Y$, although not a one-to-one exclusive point correspondence, the registration is computed as in eq. (6.4):

$$(\mathbf{q}, d) = \min \| \mathbf{x} - \mathbf{p} \|$$  \hspace{1cm} (6.4)
where $\vec{q}$ represents the EO or RO and IO parameters.

The positions of the raster data shape point set are then updated via $P = \vec{q}(P)$, where $\vec{q}$ is the transformation: the camera orientation parameters.

The point set $P$ with $N_p$ points $\vec{p}_i$ from the data shape and the model shape $X$ with $N_x$ pixels are given. The iteration is initialized by setting $P_0 = P$, $\vec{q}_0 = [X_c, Y_c, Z_c, \omega, \phi, \kappa]^\top$, the GPS/INS raw values or ROP starting values. The registration vectors are defined relative to the initial dataset $P_0$ so that the final registration represents the complete transformation. Steps a, b, c, and d are applied until convergence within a tolerance $\tau$.

1. a. Compute the closest points: $Y_k = C(P_k, X)$.

2. b. Compute the registration: $(\vec{q}_k, d_k) = Q(P_0, Y_k)$, where $Q$ is the utilized estimator

3. c. Apply the registration: $P_{k+1} = \vec{q}_k(P_0)$

4. d. Terminate the iteration when the change in meansquare error falls below a preset threshold $\tau > 0$, specifying the desired precision of the registration: $d_k - d_{k+1} < \tau$.

6.1.1 DT Convergence

A convergence theorem for the iterative algorithm is now stated and proved [17],[87]. The key ideas are that: 1) optimization generically reduces the average distance between corresponding points during each registrative iteration, and 2) the closest point determination generically reduces the distance for each point individually.
**Theorem:**

The iterative application of the distance transform algorithm always converges monotonically to a local minimum with respect to the mean-square distance objective function.

**Proof:** Given $P_k = \mathbf{p}_{ik} = \mathbf{q}_k(P_0)$ and $X$, compute the set of closest points $Y_k = \mathbf{y}_{ik}$ as prescribed above, given the internal geometric representation of $X$ and a transformation function. The mean squared error $e_k$ of that correspondence is given by eq. (6.5), the always current version of eq. (6.4):

$$e_k = \frac{1}{N_p} \sum \| \mathbf{y}_{ik} - \mathbf{p}_{ik} \|^2$$

(6.5)

The $Q$ operator is applied to obtain $\mathbf{q}_k$ and $d_k$ from that correspondence by eq. (6.6):

$$d_k = \frac{1}{N_p} \sum \| \mathbf{y}_{ik} - f(\lambda M(\mathbf{q}_{i0'})) \|^2$$

(6.6)

where $\lambda$ is the scale, $M$ is the rotation matrix based on the orientation angles $\mathbf{q}_{i0'}$ is the initial value of the camera parameters, and $f(\lambda M(\mathbf{q}_{i0'}))$ is the collinearity equation, eq.(2.12), the equation projecting the ground points to the image.

If $d_k > e_k$, then the identity transformation on the point set would yield a smaller mean square error than the registration, which cannot possibly be the case. Next, let the registration $\mathbf{q}_k$ be applied to the point set $P_0$, yielding the point set $P_{k+1}$. If the previous correspondence to the set of points $Y_k$ were maintained, then the mean
square error is still \(d_k\), that is eq.(6.7):

\[
e_k = \frac{1}{N_p} \sum \| \vec{y}_{ik} - \vec{p}_{i,k+1} \|^2 \tag{6.7}
\]

However, during the application of the subsequent closest point operator, a new point set \(Y_{k+1}\) is obtained: \(Y_{k+1} = \mathcal{C}(P_{k+1}, X)\). It is clear that eq.(6.8):

\[
\| \vec{y}_{i,k+1} - \vec{p}_{i,k+1} \| \leq \| \vec{y}_{ik} - \vec{p}_{i,k+1} \| \tag{6.8}
\]

for each \(i = 1, N_p\) because the point \(\vec{y}_{ik}\) was the closest point prior to transformation by \(\vec{q}_k\), and resides at some new distance relative to \(\vec{q}_{i,k+1}\). If \(\vec{y}_{i,k+1}\) were further from than \(\vec{y}_{ik}\), then this would directly contradict the basic operation of the \(\mathcal{C}\) closest point operator. Therefore, the mean square errors \(e_k\) and \(d_k\) must obey the following inequality eq. (6.9):

\[
0 \leq d_{k+1} \leq e_{k+1} \leq d_k \leq e_k \tag{6.9}
\]

for all \(k\). The lower bound occurs, since mean-square errors cannot be negative. Because the mean-square error sequence is non-increasing and bounded below, the algorithm as stated above must converge monotonically to a minimum value.

### 6.1.2 Camera orientation parameters from lidar only data

The camera parameters will be adjusted to the lidar data. The edge detected points from the lidar data will be projected onto the image plane, given the initial GPS/INS parameters. The distance transform will come from the edge detected
features, and the points closest to these features within a 15 pixel range will be used in the adjustment to filter out noise. The initial values are the GPS/INS values.

The ground truth was lidar data. Both the lidar data and digital images were from calibrated sources used in an actual, high accuracy survey. It should be noted here that inspired by traditional photogrammetric solutions, the terms “left image” and “right image” are used to indicate any two successive overlapping frames in a sequence of images.

The process is to create an edge map from the EDISON edge detector on the images, next, the Hough line transform is then used to find straight edges that correspond to building edges, among other features. Edges longer than 80 pixels were chosen. The edge extraction process is repeated for the lidar data. The edges extracted were usable, and were able to provide initial estimates using the proposed method. After the Hough line transform, the distance transform is applied to the Hough detected lines. This distance map will show what the closest distance is from a non-edge pixel to any edge pixel. If the closest edge pixel is known, obviously its $x, y$ coordinates are known and two separate maps are created: one map in which each non-edge pixel is given the value of the $x$ coordinate of the closest edge-pixel, and another map in which each non-edge pixel is given the value of the $y$ coordinate of the closest edge-pixel. Since the lidar edge detector still contains noise, the distance function search was limited to 15 pixels, so that noisy edges do not unduly influence the orientation. Once the coarse registration is done using the Hough transform, the image was used for refining the orientation parameters, using the closed EDISON detected contours, matched to the lidar detected edges. After this refined adjustment, a further, more refined adjustment is performed using the Canny edge detector and the
GIS data. The Canny edge detector is useful for finding individual linear features, not just closed contours, like street markings and curbs. These linear features are then compared with GIS data containing curbs or street markings. The curb data, if fused with the lidar data, will contain elevation data, and thus may be used in the adjustment process.

6.1.3 Extraction of building outline from lidar data

Many fine results have been obtained for building extraction from lidar data [113] [114]. The approach in [95] is used, since it is simple, fast and accurate for large buildings. It is possible to separate buildings candidate points from lidar data using the assumption that buildings are higher than surrounding area. In this study, the local maxima filter is used to extract building candidate points.

As mentioned in section(4.2.1), it is difficult to derive surface semantic information from lidar, so the lidar data has to be filtered for vegetation and other noise; such as areas with high elevation variability before edge extraction. Fig(6.1) shows the high variability areas: vegetation, parking lot as black, and constant slope areas as white. This filter was based on the Harris corner detector [67]. The vegetation and cars are shown as high variability elevation data and can then be filtered out. Fig(6.2) shows the final result, with the lidar edges projected on to image 425.

Given multimodal datasets of a scene, based on the sensors and filters and the knowledge of the human operator, the extracted surface features may look different, however, these are all different representations of the same real world surface [148]. Following this conjecture, consider the discontinuities in both the lidar and image. The discontinuities in lidar data represent elevation differences of the surface, such
as between the facades of a building. Depending on the view point, a subset of these 3D discontinuities manifest themselves in the images as edges observed due to parallax. Note that edges extracted from an image contain such edges, and include additional edges observed due to texture changes (see Figure 6.2). In order to detect discontinuities from the lidar data, the edges are detected in lidar elevation data. As seen in Figure 6.2, the edges in the image and the projected lidar data correspond to the depth discontinuities.

6.1.4 Harris corner detection

The Harris corner detector basic idea is to find points where two edges meet—i.e., high gradient in two directions. The detector is based on the pixel values. If the pixel values represent elevation, variable elevation points can be located.

For a 2-dimensional image, given by \( I \), consider an image patch over an area \((u, v)\) and shifting it by \((x, y)\). The weighted “sum of squared differences” (SSD) between these two patches, denoted \( S \), is given by:

\[
S(x, y) = \sum_u \sum_v w(u, v) \left( I(u + x, v + y) - I(u, v) \right)^2
\]

\( I(u + x, v + y) \) can be approximated by a Taylor series. Let \( I_x \) and \( I_y \) be the partial derivatives of \( I \), such that :

\[
I(u + x, v + y) \approx I(u, v) + I_x(u, v)x + I_y(u, v)y.
\]

This produces the approximation :

\[
S(x, y) \approx \sum_u \sum_v w(u, v) \left( I_x(u, v)x + I_y(u, v)y \right)^2,
\]

which can be written in matrix form: \( S(x, y) \approx \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix}, \) where:

\[
A = \sum_u \sum_v w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}
\]
This matrix is a Harris matrix, and angle brackets denote averaging, i.e. summation over \((u,v)\). If a circular window or circularly weighted window, such as a Gaussian function is used, then the response will be isotropic.

A corner, or in general an interest point, is characterized by a large variation of \(S\) in all directions of the vector \((x\ y)\). By analyzing the eigenvalues of \(A\), this characterization can be expressed in the following way: \(A\) should have two “large” eigenvalues for an interest point. Based on the magnitudes of the eigenvalues, the following inferences can be made based on this argument: if \(\lambda_1 \approx 0\) and \(\lambda_2 \approx 0\) then this pixel \((x, y)\) has no features of interest, if \(\lambda_1 \approx 0\) and \(\lambda_2\) has some large positive value, then an edge is found. If \(\lambda_1\) and \(\lambda_2\) have large positive values, then a corner is found.

### 6.1.5 Hough transform

To reduce the clutter in the edge extracted images, and only to use straight edges that are longer than cars, the Hough transform was used to search for longer edges in the images. The Hough transform (HT) is a general technique for identifying the locations and orientations of certain types of features in a digital image. Developed by Paul Hough in 1962 [73] and patented by IBM, the transform consists of parameterizing a description of a feature at any given location in the original image’s space.

The basic principle of the Hough transform was to switch the roles of parameters and spatial variables [62]. The Hough line transform attempts to describe lines that match straight edges in a two-dimensional image. To illustrate this approach, consider the following example. After edge detection, an image is searched for lines, (see fig.6.3). A line can be defined by the Hesse normal form:
where $\phi$ is the angle of inclination of the normal and $p$ is the length of the normal. The normal is defined to be the shortest segment between the line in question and the origin. This equation specifies a line passing through any edge point $(x, y)$ that is perpendicular to the polar line drawn from the origin to $\rho, \theta$ in polar space, or i.e. $\rho \cos(\theta), \rho \sin(\theta)$ in cartesian coordinates. For each edge point $(x, y)$ on the same line, $\rho$ and $\theta$ are constant. For any given point, an infinite amount of lines pass through that point, each with different values for $\rho$ and $\theta$. By iterating through all possible angles for $\theta$, there is a unique solution for $\rho$.

To generate the Hough transform for matching lines in the image, a particular discretization is chosen for the lines (e.g., $10^\circ$ for $\theta$), and then iterated through the angles defined by that value. For each angle $\theta$, solve for the Hesse normal form in eq. (6.10), and then increment the value located at $(\rho, \theta)$ in an “accumulator” array. This process can be thought of as a “vote” by the point $(x, y)$ for the line defined by $(\rho, \theta)$, and variations on how votes are generated exist for making the transform more robust.

After the Hough line transform, the distance transform is applied to the Hough detected lines. This distance map will show what the closest distance is from a non-edge pixel to any edge pixel. If the closest edge pixel is known, obviously its $x, y$ coordinates are known and two separate maps are created: one map in which each a non-edge pixel is given the value of the $x$ coordinate of the closest edge-pixel, and another map in which each non-edge pixel is given the value of the $y$ coordinate of the closest edge-pixel. This allows a comparison to the projected lidar or GIS data: if
the projected point falls on a non-edge pixel, the coordinates of the closest edge pixel are known from the distance transform, and since the pixel coordinates of the non-edge pixel are known, onto which the projected point falls, this allow for setting up a distance function for minimization. Since the Canny edge detector and the Hough line transform still contains noise, the distance function is limited to 15 pixels, so that noisy edges do not unduly influence the orientation.

6.1.6 Relative orientation

Many of the traditional conjugate point finding operators are based on auto-correlation of the image areas, like the Moravec or Förstner operator, or using a descriptor that is a 3-D histogram of gradient location and orientation, like the SIFT operator [133] [100]. The proposed method will use the DT to find conjugate points from the data projected into object space. Since this method needs close approximations, the initial parameters will be calculated by SIFT. The coordinates from SIFT for matched points will be processed to create the surface points in the RO space, and gain initial approximates for the RO parameters, (see fig 6.5). The process after this is the same as before; use these initial parameters in the same manner as for EO: project the 3-D coordinates to the right image, since the left image parameters are fixed, and carry out an adjustment. For computing ROPs, the setup described in [182] is used:

1. set left image orientation angles to $0^\circ$

2. set left image perspective center coordinates to $P_{lc} = (0, 0, f)$

3. set right image $x$ coordinate to the photo base
4. set right image initial values as $\omega_r = 0, \phi_r = 0, \kappa_r = 0, Z_{rc} = f$

The process is shown in fig.(6.4).

After initial parameters are recovered by SIFT, the EDISON extracted edges are used as a refinement, and finally, the Canny detected edges are used to find the final parameters. This method is the traditional dependent relative orientation, leaving five elements of the right image orientation parameters to be solved. The traditional method is to find conjugate points along the Gruber areas of the two images, and run a least squares adjustment.

6.2 Results

6.2.1 Camera orientation from lidar only and from GIS and lidar

As can be seen from the results, the adjusted points fall close to the extracted edges, fig.(6.7) for lidar only and fig.(6.8) for GIS, which is closer than the raw GPS/INS values. The GIS dataset was altered for some of the experiments. In order to show how elevation affects the results, the GIS dataset was altered to add generic elevations of 270 m, 275 m and 280 m to the California GIS data. The average elevation of the building rooftops for that area is $\sim 275$ m. A lower and higher elevation was used to study the effect. Gaussian noise was also added to the data, to study the robustness of the proposed method under noisy conditions, see (fig. 6.6). The average displacement of the noise was 0 m in each axis direction with a standard deviation of 1 m.
The least squares equation shown in section 4.3.1 will be altered with the Levenberg-Marquardt dampener:

$$\Delta p = -(J^TPJ + \lambda diag(J^TPJ))^{-1}J^TPf(p_0),$$

(6.11)

which was explained earlier in section 4.3.1.

For quantitative comparison, two sets of reviews are provided. The first set introduces the estimated parameters and the resulting standard deviations; while the second review measures the quality of surface generation from the stereo-compilation using the two images against the ground truth lidar data. The tabulated initial and estimated camera parameters for the California study site are in Tables 6.1 - 6.3. These tables are the LSE results using a dampener value of 1 after 15 iterations. Using a dampener value of 0 resulted in higher standard deviations, if there was any convergence at all. Using a higher dampener value resulted in much slower convergence. Regarding the standard deviations, for the image 425, the convergence stopped after six iterations, and the results did not improve. However, for image 426 and 427, the solution was improving even after 10 iterations. Using the proposed method, with regards to the focal length and principal point values, there is a noticeable tendency of the focal length to move to a value of $f < 14$ mm, and the $x_0$ principal point value to be slightly larger than 0, the $y_0$ principal point value to be slightly less than 0. The manual method has mixed results: sometimes $f > 14$, sometimes $f < 14$, and the principal point coordinates also are variable. A comparison of some of the points, with the number of points reduced for clarity, projected to the edge detection results
of the image 425 can be seen in fig.(6.7) and fig.(6.8). The MLE implementation gave almost identical results, except for significant digits.

The second quantitative comparison is facilitated by first estimating the camera parameters for each image independently from the \( \text{<image,lidar>} \) pair and then using the estimated parameters to perform the stereo compilation between manually selected 32 corresponding points. The distances \( d_i \) between the recovered 3D points, \( \tilde{X}_i^{\text{intersect}} \), and the corresponding lidar points, \( X_i^{\text{lidar}} \), serve as the comparison criteria. In Tables 6.4 and 6.6, the mean, \( \mu_d \), and the standard deviation, \( \sigma_d \), are tabulated, for the distances computed using three different camera parameter estimation methods: 1) manual, 2) \( \text{<image,lidar>} \) pair and 3) \( \text{<image,GIS>} \) pair.

Table 6.5 tabulates the quantitative comparison results for The Ohio State University study site for all three methods. The variance of unit weight \( \sigma_0 \) parameter for manual, lidar-based and GIS-based adjustments, respectively computed as 0.0312, 0.3700 and 0.2475. Similar to the California results, the results from the proposed approach is better than the manual approach. Note that the intrinsic camera parameters comparably change more than results for the California study site.

Overall, note that while better than the manual approach, the OSU site results are not as good as the California site. This observation is primarily due to the extreme scale difference between the imagery used for both sites, as well as the density of structures in the environment. For instance, the buildings outside the OSU campus area are only a few feet apart making even manual selection of control points difficult. This observation extends to the lidar and GIS data where all buildings generate a repetitive pattern and distinguishing them from one another is very hard. This observation is manifested in the 3D recovery results tabulated in Table 6.6. While
<table>
<thead>
<tr>
<th>Variable</th>
<th>$X_0$ [m]</th>
<th>$Y_0$ [m]</th>
<th>$Z_0$ [m]</th>
<th>$\sigma^X$ [mm]</th>
<th>$\sigma^Y$ [mm]</th>
<th>$\sigma^Z$ [mm]</th>
<th>$\sigma^\xi$ [mm]</th>
<th>$\sigma^\eta$ [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>image 1: 473518.8157</td>
<td>709448.7488</td>
<td>650.4514</td>
<td>-2.521</td>
<td>-3.622</td>
<td>-3.622</td>
<td>-4.640</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>image 2: 473518.8157</td>
<td>709448.7632</td>
<td>651.9042</td>
<td>-2.541</td>
<td>-3.570</td>
<td>-4.604</td>
<td>0.002</td>
<td>-0.003</td>
<td></td>
</tr>
<tr>
<td>$\sigma$</td>
<td>7.136 x 10^{-10}</td>
<td>0.000</td>
<td>2.459 x 10^{-10}</td>
<td>3.130 x 10^{-10}</td>
<td>2.380 x 10^{-10}</td>
<td>1.104 x 10^{-9}</td>
<td>2.364 x 10^{-9}</td>
<td></td>
</tr>
<tr>
<td>$\sigma_0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>image 1: 473519.9115</td>
<td>709448.8612</td>
<td>650.4416</td>
<td>-2.550</td>
<td>-3.609</td>
<td>-4.640</td>
<td>0.001</td>
<td>0.004</td>
<td></td>
</tr>
<tr>
<td>image 2: 473519.9115</td>
<td>709448.5414</td>
<td>651.5534</td>
<td>-2.554</td>
<td>-3.661</td>
<td>-4.640</td>
<td>0.002</td>
<td>-0.011</td>
<td></td>
</tr>
<tr>
<td>image 1: 473519.1613</td>
<td>709448.5414</td>
<td>651.5534</td>
<td>-2.564</td>
<td>-3.661</td>
<td>-4.640</td>
<td>0.002</td>
<td>-0.011</td>
<td></td>
</tr>
<tr>
<td>image 2: 473519.1613</td>
<td>709448.5715</td>
<td>647.7725</td>
<td>-2.520</td>
<td>-3.658</td>
<td>-4.640</td>
<td>0.005</td>
<td>-0.006</td>
<td></td>
</tr>
<tr>
<td>image 1: 473519.1258</td>
<td>709448.7075</td>
<td>647.7725</td>
<td>-2.520</td>
<td>-3.658</td>
<td>-4.640</td>
<td>0.005</td>
<td>-0.006</td>
<td></td>
</tr>
<tr>
<td>image 2: 473519.1258</td>
<td>709448.8612</td>
<td>650.4416</td>
<td>-2.550</td>
<td>-3.609</td>
<td>-4.640</td>
<td>0.001</td>
<td>0.004</td>
<td></td>
</tr>
<tr>
<td>image 1: 473519.3614</td>
<td>709448.5414</td>
<td>651.5534</td>
<td>-2.564</td>
<td>-3.661</td>
<td>-4.640</td>
<td>0.002</td>
<td>-0.011</td>
<td></td>
</tr>
<tr>
<td>image 2: 473519.3614</td>
<td>709448.6316</td>
<td>648.2238</td>
<td>-2.520</td>
<td>-3.658</td>
<td>-4.640</td>
<td>0.005</td>
<td>-0.006</td>
<td></td>
</tr>
<tr>
<td>image 1: 473519.5510</td>
<td>709448.5332</td>
<td>650.1912</td>
<td>-2.552</td>
<td>-3.713</td>
<td>-4.640</td>
<td>0.006</td>
<td>-0.010</td>
<td></td>
</tr>
<tr>
<td>image 2: 473519.5510</td>
<td>709448.5297</td>
<td>651.0849</td>
<td>-2.555</td>
<td>-3.669</td>
<td>-4.640</td>
<td>0.002</td>
<td>-0.011</td>
<td></td>
</tr>
<tr>
<td>$\sigma$</td>
<td>4.010 x 10^{-10}</td>
<td>8.113 x 10^{-10}</td>
<td>5.548 x 10^{-10}</td>
<td>1.212 x 10^{-9}</td>
<td>9.681 x 10^{-10}</td>
<td>1.769 x 10^{-9}</td>
<td>3.089 x 10^{-9}</td>
<td></td>
</tr>
</tbody>
</table>
Table 6.2: Image 426 solutions

<table>
<thead>
<tr>
<th>Variable</th>
<th>(X) [m]</th>
<th>(Y) [m]</th>
<th>(Z) [m]</th>
<th>(\chi)</th>
<th>(\nu)</th>
<th>(\tau)</th>
<th>(\sigma)</th>
<th>(\alpha)</th>
<th>(\beta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,397,7289</td>
<td>654.0002</td>
<td>-1.6816</td>
<td>-3.3216</td>
<td>-141.5624</td>
<td>14.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,397,9567</td>
<td>654.5803</td>
<td>-1.7180</td>
<td>-3.4003</td>
<td>-141.7952</td>
<td>14.0320</td>
<td>0.0134</td>
<td>-0.0228</td>
</tr>
<tr>
<td>GPS/IMU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.72E-08</td>
<td>5.38E-08</td>
<td>5.40E-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,397,5275</td>
<td>654.0864</td>
<td>-1.6925</td>
<td>-4.0400</td>
<td>-141.9908</td>
<td>14.0963</td>
<td>0.0057</td>
<td>-0.0165</td>
</tr>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,396,9044</td>
<td>654.5617</td>
<td>-1.7243</td>
<td>-3.3913</td>
<td>-141.8520</td>
<td>13.9872</td>
<td>0.0118</td>
<td>-0.0207</td>
</tr>
<tr>
<td>GIS lidar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.72E-08</td>
<td>5.38E-08</td>
<td>5.40E-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,397,1579</td>
<td>650.4299</td>
<td>-1.6945</td>
<td>-4.3991</td>
<td>-141.9308</td>
<td>14.1264</td>
<td>0.0118</td>
<td>-0.0170</td>
</tr>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,397,0876</td>
<td>651.0601</td>
<td>-1.7026</td>
<td>-3.4122</td>
<td>-141.9265</td>
<td>13.9883</td>
<td>0.0054</td>
<td>-0.0083</td>
</tr>
<tr>
<td>GIS 270m elev.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.72E-08</td>
<td>5.38E-08</td>
<td>5.40E-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,397,8891</td>
<td>650.6239</td>
<td>-1.6969</td>
<td>-4.6023</td>
<td>-141.9265</td>
<td>13.9883</td>
<td>0.0054</td>
<td>-0.0083</td>
</tr>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,397,0075</td>
<td>655.2050</td>
<td>-1.7088</td>
<td>-3.4199</td>
<td>-141.9265</td>
<td>13.9090</td>
<td>0.0014</td>
<td>-0.0067</td>
</tr>
<tr>
<td>GIS 275m elev.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.72E-08</td>
<td>5.38E-08</td>
<td>5.40E-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,396,9350</td>
<td>654.5320</td>
<td>-1.7198</td>
<td>-3.3880</td>
<td>-141.8615</td>
<td>13.9880</td>
<td>0.0120</td>
<td>-0.0203</td>
</tr>
<tr>
<td>Image 426</td>
<td>473,574</td>
<td>769,396,6648</td>
<td>654.5320</td>
<td>-1.7198</td>
<td>-3.3880</td>
<td>-141.8615</td>
<td>13.9880</td>
<td>0.0120</td>
<td>-0.0203</td>
</tr>
<tr>
<td>GIS lidar noise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.72E-08</td>
<td>5.38E-08</td>
<td>5.40E-04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 6.3: Image 427 solutions

<table>
<thead>
<tr>
<th>Variable</th>
<th>( X )</th>
<th>( Y )</th>
<th>( Z )</th>
<th>( \sigma )</th>
<th>( \sigma_1 )</th>
<th>( \sigma_2 )</th>
<th>( \sigma_3 )</th>
<th>( \omega_1 )</th>
<th>( \omega_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>471434.4177</td>
<td>709336.3747</td>
<td>657.2850</td>
<td>0.3537</td>
<td>-1.2340</td>
<td>-141.9719</td>
<td>14.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Manual</td>
<td>471434.5896</td>
<td>709336.6433</td>
<td>657.9386</td>
<td>0.3123</td>
<td>-1.3882</td>
<td>-142.0123</td>
<td>14.0513</td>
<td>0.0078</td>
<td>-0.0216</td>
</tr>
<tr>
<td>Image 2</td>
<td>473634.6272</td>
<td>709346.5715</td>
<td>657.3953</td>
<td>0.3853</td>
<td>-1.3399</td>
<td>-142.3707</td>
<td>14.1242</td>
<td>0.0041</td>
<td>-0.0268</td>
</tr>
<tr>
<td>Lidar</td>
<td>473635.0975</td>
<td>709335.9525</td>
<td>657.2390</td>
<td>0.3893</td>
<td>-1.3176</td>
<td>-142.2860</td>
<td>14.0055</td>
<td>0.0125</td>
<td>-0.0237</td>
</tr>
<tr>
<td>Image 3</td>
<td>473635.2175</td>
<td>709335.2472</td>
<td>653.7771</td>
<td>0.3658</td>
<td>-1.3221</td>
<td>-142.2295</td>
<td>14.1243</td>
<td>0.0147</td>
<td>-0.0286</td>
</tr>
<tr>
<td>GIS lidar</td>
<td>473635.5960</td>
<td>709335.5900</td>
<td>651.8596</td>
<td>0.3865</td>
<td>-1.3879</td>
<td>-142.2026</td>
<td>13.9979</td>
<td>0.0070</td>
<td>-0.0046</td>
</tr>
<tr>
<td>Image 4</td>
<td>473635.6306</td>
<td>709346.0610</td>
<td>655.6956</td>
<td>0.4055</td>
<td>-1.3933</td>
<td>-142.1140</td>
<td>13.9367</td>
<td>0.0039</td>
<td>0.0038</td>
</tr>
<tr>
<td>GIS lidar noise</td>
<td>473635.0519</td>
<td>709345.9760</td>
<td>657.2156</td>
<td>0.4018</td>
<td>-1.3115</td>
<td>-142.2797</td>
<td>14.0064</td>
<td>0.0122</td>
<td>-0.0233</td>
</tr>
</tbody>
</table>
Table 6.4: Mean distances computed in meters for X, Y and Z coordinates between lidar points and recovered 3D points for image 425 and 426 of the California study site. The recovered points are estimated by intersecting the rays from corresponding points between two images whose camera parameters are estimated using the proposed method.

<table>
<thead>
<tr>
<th></th>
<th>Initial</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>µ&lt;sub&gt;d&lt;/sub&gt;</td>
<td>-0.349</td>
<td>-0.165</td>
</tr>
<tr>
<td>σ&lt;sub&gt;d&lt;/sub&gt;</td>
<td>0.203</td>
<td>0.322</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>&lt;image,lidar&gt;</th>
<th>&lt;image,GIS&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>µ&lt;sub&gt;d&lt;/sub&gt;</td>
<td>0.312</td>
<td>-0.258</td>
</tr>
<tr>
<td>σ&lt;sub&gt;d&lt;/sub&gt;</td>
<td>0.289</td>
<td>0.364</td>
</tr>
</tbody>
</table>

not better than the manual approach as it is in the results shown for the California site, the estimated 3D, when GIS data is used, is still comparable to the manual results and improves the results computed using the initial camera parameters. The result estimated using the lidar data, however, is not as good due to fusing sparse lidar data with a very low scale image whose edges do not align well after the extraction process.

The closest points/pixels on the edge detected image were considered the conjugate points of the projected laser points. Back projecting the closest edge data using the inverse collinearity equation, taking data from 24 points, the results in Table 6.7 for images 425-427 describe the back projected data for the images in terms of mean difference and standard deviation from lidar data, using the elevation of the closest laser point as a benchmark. The 24 points were distributed equally, and 12 of these were on well defined areas, like building or curb corners that are available on the GIS and used in the proposed adjustment, and 12 in non-well defined areas, like
Table 6.5: Solutions for (a) the principal point offset and focal length all in millimeters; (b) perspective center coordinates in meters; and (c) camera orientation in degrees. The results are generated for the Ohio State University study site.

<table>
<thead>
<tr>
<th>Method</th>
<th>$f$</th>
<th>$x_0$</th>
<th>$y_0$</th>
<th>$\sigma_f$</th>
<th>$\sigma_{x_0}$</th>
<th>$\sigma_{y_0}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>152.152</td>
<td>-0.015</td>
<td>0.012</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Manual</td>
<td>152.224</td>
<td>0.0134</td>
<td>-0.0228</td>
<td>0.0033</td>
<td>0.0054</td>
<td>0.001</td>
</tr>
<tr>
<td>&lt;image,lidar&gt;</td>
<td>154.785</td>
<td>-0.7121</td>
<td>0.9789</td>
<td>0.3106</td>
<td>0.3164</td>
<td>0.0321</td>
</tr>
<tr>
<td>&lt;image,GIS&gt;</td>
<td>151.902</td>
<td>0.042</td>
<td>0.272</td>
<td>0.018</td>
<td>0.012</td>
<td>0.015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>$X_c$</th>
<th>$Y_c$</th>
<th>$Z_c$</th>
<th>$\sigma_{X_c}$</th>
<th>$\sigma_{Y_c}$</th>
<th>$\sigma_{Z_c}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>1825475.784</td>
<td>732332.075</td>
<td>7034.082</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Manual</td>
<td>1825476.557</td>
<td>732332.5923</td>
<td>7035.9460</td>
<td>1.938</td>
<td>3.741</td>
<td>1.078</td>
</tr>
<tr>
<td>&lt;image,lidar&gt;</td>
<td>1825476.561</td>
<td>732331.9663</td>
<td>7034.282</td>
<td>4.2108</td>
<td>6.8547</td>
<td>7.3994</td>
</tr>
<tr>
<td>&lt;image,GIS&gt;</td>
<td>1825475.9126</td>
<td>732332.4860</td>
<td>7033.9606</td>
<td>0.6935</td>
<td>0.5656</td>
<td>0.5350</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>$\omega$</th>
<th>$\phi$</th>
<th>$\kappa$</th>
<th>$\sigma_\omega$</th>
<th>$\sigma_\phi$</th>
<th>$\sigma_\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>-1.4872</td>
<td>0.2320</td>
<td>89.5131</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Manual</td>
<td>-1.4916</td>
<td>0.2266</td>
<td>89.5133</td>
<td>0.0698</td>
<td>0.0569</td>
<td>0.0539</td>
</tr>
<tr>
<td>&lt;image,lidar&gt;</td>
<td>-1.5181</td>
<td>0.4619</td>
<td>89.01172</td>
<td>0.0201</td>
<td>0.3580</td>
<td>0.0267</td>
</tr>
<tr>
<td>&lt;image,GIS&gt;</td>
<td>-1.4911</td>
<td>0.2265</td>
<td>89.5133</td>
<td>0.002</td>
<td>0.0488</td>
<td>0.0306</td>
</tr>
</tbody>
</table>

the centroid of road intersections, not used in the proposed adjustment. The results show that these points are within a half meter of the expected result. However, this is somewhat misleading, as for the well defined points that are on features with coordinates, the difference is less than 10 cm. The problem is the areas that have no control. If more GIS features are available for adjustment, maybe road markings, then the results may improve.

For visual comparison, the results were draped over the lidar data, created from a TIN representation using Arcview 3-D Analyst (see fig. 6.9).
Table 6.6: Mean distances computed in meters for $X$, $Y$ and $Z$ coordinates between lidar points and recovered 3D points for the OSU study site. The table includes results using initial parameters and adjusted parameters for manual and proposed approach.

<table>
<thead>
<tr>
<th></th>
<th>Initial</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X$</td>
<td>$Y$</td>
</tr>
<tr>
<td>$\mu_d$</td>
<td>2.586</td>
<td>-2.219</td>
</tr>
<tr>
<td>$\sigma_d$</td>
<td>1.829</td>
<td>2.372</td>
</tr>
</tbody>
</table>

Table 6.7: Mean distances computed in meters for $X$, $Y$ and $Z$ coordinates between lidar points and recovered 3D points for the California study site, using proposed approach with GIS. The table includes results using adjusted parameters, using points matched to the exact lidar elevation. (a) Image 425 (b) Image 426 (c) Image 427.

<table>
<thead>
<tr>
<th>Coord.</th>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean difference meter</td>
<td>-0.229</td>
<td>-0.183</td>
<td>0</td>
</tr>
<tr>
<td>Std. dev. meter</td>
<td>0.185</td>
<td>0.321</td>
<td>0</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>Coord.</th>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean difference meter</td>
<td>0.151</td>
<td>0.237</td>
<td>0</td>
</tr>
<tr>
<td>Std. dev. meter</td>
<td>0.215</td>
<td>0.355</td>
<td>0</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th>Coord.</th>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean difference meter</td>
<td>0.323</td>
<td>-0.363</td>
<td>0</td>
</tr>
<tr>
<td>Std. dev. meter</td>
<td>0.453</td>
<td>0.369</td>
<td>0</td>
</tr>
</tbody>
</table>

(c)

Please see Appendix A for further draping results of Image 425, and for all results of Images 426 and 427.
6.2.2 Bundle adjustment of images

To obtain consistent data for a series of images, bundle adjustment was performed on the three images. The point-based methodology involves finding GCPs on the images, and tie points on the images, so the images remain correctly oriented with respect to each other and to the ground. The actual ground control will be the lidar or GIS points. Instead of using tie points, the numerous lidar points will represent the tie points, thus eliminating the need to search for individual tie points or features on the individual images. This way, all three images have the same tie features, orienting the images correctly with respect to each other.

While these solutions are close to the individual image orientations, they are not the same. Still, the focal length and the principal point coordinate movement direction matched that of the individual results. Similar results are acquired by MLE. Please see Appendix B for a bar graph representation of the results.

6.2.3 Relative orientation

The edgemaps can be compared using DT after SIFT for a more accurate calculation of ROPs. However, the short lines may be mistaken for noise, and have to be filtered out for accurate displacement. The edges after displacement are shown in fig. 6.11, with different shades representing the two edgemaps. As can be seen, the edges complement each other quite well. For these experiments, image 425 is used as the right image, image 426 is used as the left image for model 1, and image 426 is used as the right image and image 427 is used as the left image for model 2.

The individual DT matching was not as accurate as manual matching, but as stated in [62] and [146], quantity compensates for quality, thus returning a similar
Table 6.8: Bundle Adjustment solutions

<table>
<thead>
<tr>
<th>Image</th>
<th>$x_1$ [m]</th>
<th>$y_1$ [m]</th>
<th>$z_1$ [m]</th>
<th>$\Delta x_1$</th>
<th>$\Delta y_1$</th>
<th>$\Delta z_1$</th>
<th>$\sigma_{\Delta x_1}$</th>
<th>$\sigma_{\Delta y_1}$</th>
<th>$\sigma_{\Delta z_1}$</th>
<th>$\sigma_{x_1}$</th>
<th>$\sigma_{y_1}$</th>
<th>$\sigma_{z_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame 1</td>
<td>437519.6378</td>
<td>7109.4938</td>
<td>650.4454</td>
<td>-2.5986</td>
<td>3.5754</td>
<td>-140.9549</td>
<td>0.0015</td>
<td>0.0002</td>
<td>0.0009</td>
<td>0.0211</td>
<td>0.0002</td>
<td>0.0009</td>
</tr>
<tr>
<td>Frame 2</td>
<td>437513.6378</td>
<td>7109.4938</td>
<td>650.4454</td>
<td>-1.7941</td>
<td>7.8020</td>
<td>-134.5830</td>
<td>13.7238</td>
<td>-0.0094</td>
<td>0.0211</td>
<td>0.0002</td>
<td>0.0009</td>
<td>0.0211</td>
</tr>
<tr>
<td>Frame 3</td>
<td>437511.6378</td>
<td>7109.4938</td>
<td>650.4454</td>
<td>-1.7845</td>
<td>3.8567</td>
<td>-134.5830</td>
<td>13.7238</td>
<td>-0.0094</td>
<td>0.0211</td>
<td>0.0002</td>
<td>0.0009</td>
<td>0.0211</td>
</tr>
<tr>
<td>Frame 4</td>
<td>437510.6378</td>
<td>7109.4938</td>
<td>650.4454</td>
<td>-1.0111</td>
<td>4.1181</td>
<td>-134.5830</td>
<td>13.7238</td>
<td>-0.0094</td>
<td>0.0211</td>
<td>0.0002</td>
<td>0.0009</td>
<td>0.0211</td>
</tr>
<tr>
<td>Frame 5</td>
<td>437511.6378</td>
<td>7109.4938</td>
<td>650.4454</td>
<td>-1.0554</td>
<td>3.5833</td>
<td>-134.5830</td>
<td>13.7238</td>
<td>-0.0094</td>
<td>0.0211</td>
<td>0.0002</td>
<td>0.0009</td>
<td>0.0211</td>
</tr>
</tbody>
</table>

Adjustment Variables:

- $x_1$: Horizontal position
- $y_1$: Vertical position
- $z_1$: Height
- $\Delta x_1$, $\Delta y_1$, $\Delta z_1$: Adjusted changes
- $\sigma_{\Delta x_1}$, $\sigma_{\Delta y_1}$, $\sigma_{\Delta z_1}$: Standard deviations
- $\sigma_{x_1}$, $\sigma_{y_1}$, $\sigma_{z_1}$: Standard deviations of adjusted values

Bundle adjustment results for different scenarios:

- Manual adjustment
- Adjusted GIS lidar
- Adjusted GIS lidar, 270m elevation
- Adjusted GIS lidar, 280m elevation
- Adjusted GIS lidar, noise
Table 6.9: Estimated relative orientation parameters and their standard deviations for manual, SIFT based and proposed approaches for Model 1. The unit of reported values are in millimeters. Boldface indicates best results with least deviation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Manual</th>
<th>SIFT</th>
<th>Proposed</th>
<th>$\sigma_{\text{manual}}$</th>
<th>$\sigma_{\text{sift}}$</th>
<th>$\sigma_{\text{proposed}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$</td>
<td>0.7811</td>
<td>0.9692</td>
<td>0.9640</td>
<td><strong>0.0045</strong></td>
<td>0.1014</td>
<td>0.0094</td>
</tr>
<tr>
<td>$\phi$</td>
<td>-0.3947</td>
<td>-0.4962</td>
<td>-0.5512</td>
<td><strong>0.0036</strong></td>
<td>0.0594</td>
<td>0.0062</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>0.5834</td>
<td>0.6562</td>
<td>0.6680</td>
<td>0.0078</td>
<td>0.0285</td>
<td>0.0029</td>
</tr>
<tr>
<td>$X_c$</td>
<td>0.5039</td>
<td>0.5159</td>
<td>0.5168</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$Y_c$</td>
<td>-3.5058</td>
<td>-3.6485</td>
<td>-3.6183</td>
<td><strong>0.0091</strong></td>
<td>0.1568</td>
<td>0.0211</td>
</tr>
<tr>
<td>$Z_c$</td>
<td>13.9116</td>
<td>14.0245</td>
<td>14.0470</td>
<td><strong>0.0053</strong></td>
<td>0.0064</td>
<td>0.0061</td>
</tr>
<tr>
<td>$f$</td>
<td>13.9936</td>
<td>14.0413</td>
<td>14.0333</td>
<td><strong>0.0033</strong></td>
<td>0.0053</td>
<td>0.0041</td>
</tr>
<tr>
<td>$x_0$</td>
<td>0.0042</td>
<td>0.0099</td>
<td>0.0044</td>
<td>0.0371</td>
<td>0.0037</td>
<td><strong>0.0025</strong></td>
</tr>
<tr>
<td>$y_0$</td>
<td>0.0014</td>
<td>0.0056</td>
<td>0.0053</td>
<td>0.0488</td>
<td>0.0063</td>
<td><strong>0.0019</strong></td>
</tr>
</tbody>
</table>

Table 6.10: Model 1 photo coordinate residuals for left and right images from (a) SIFT matching, (b) DT matching, (c) manual matching:

<table>
<thead>
<tr>
<th>point</th>
<th>x left res.</th>
<th>y left res.</th>
<th>x right res.</th>
<th>y right res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS [mm]</td>
<td>0.0006</td>
<td>0.0001</td>
<td>0.0006</td>
<td>0.0001</td>
</tr>
<tr>
<td>(a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>point</td>
<td>x left res.</td>
<td>y left res.</td>
<td>x right res.</td>
<td>y right res.</td>
</tr>
<tr>
<td>RMS [mm]</td>
<td>0.0006</td>
<td>0.0010</td>
<td>0.0006</td>
<td>0.0010</td>
</tr>
<tr>
<td>(b)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>point</td>
<td>x left res.</td>
<td>y left res.</td>
<td>x right res.</td>
<td>y right res.</td>
</tr>
<tr>
<td>RMS [mm]</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0002</td>
<td>0.0001</td>
</tr>
<tr>
<td>(c)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

matching quality as a final result. The DT matching was able to match more points in less time across the overlap of the images, compensating for the lower individual quality. Interestingly, the proposed method, in both relative and exterior orientation, increased the $x_0$ coordinate of the principal point offset, while manual matching decreased it for the exterior orientation. For model 2, the manual and SIFT results may be unreliable, due to the high standard error of unit weight. This could be due to:
Table 6.11: Estimated relative orientation parameters and their standard deviations for manual, SIFT based and proposed approaches for Model 2. The unit of reported values are in millimeters. Boldface indicates best results with least deviation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Manual</th>
<th>SIFT</th>
<th>Proposed</th>
<th>$\sigma_{\text{manual}}$</th>
<th>$\sigma_{\text{sift}}$</th>
<th>$\sigma_{\text{proposed}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$</td>
<td>0.7217</td>
<td>0.7749</td>
<td>0.8005</td>
<td>36.4505</td>
<td>1.4868</td>
<td><strong>0.1303</strong></td>
</tr>
<tr>
<td>$\phi$</td>
<td>-0.1259</td>
<td>-0.2084</td>
<td>-0.2382</td>
<td>9.1098</td>
<td>0.6462</td>
<td><strong>0.0641</strong></td>
</tr>
<tr>
<td>$\kappa$</td>
<td>0.2954</td>
<td>0.3494</td>
<td>0.3439</td>
<td>6.9860</td>
<td>0.3103</td>
<td><strong>0.0302</strong></td>
</tr>
<tr>
<td>$X_c$</td>
<td>0.2814</td>
<td>0.2772</td>
<td>0.2759</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$Y_c$</td>
<td>-3.4004</td>
<td>-3.0305</td>
<td>-3.1973</td>
<td>5.6467</td>
<td>1.7882</td>
<td><strong>0.2513</strong></td>
</tr>
<tr>
<td>$Z_c$</td>
<td>14.0750</td>
<td>14.0339</td>
<td>14.0028</td>
<td>1.4543</td>
<td>0.0514</td>
<td><strong>0.0073</strong></td>
</tr>
<tr>
<td>$f$</td>
<td>13.9125</td>
<td>14.0083</td>
<td>14.0012</td>
<td>0.0338</td>
<td>0.0053</td>
<td><strong>0.0041</strong></td>
</tr>
<tr>
<td>$x_0$</td>
<td>0.0088</td>
<td>0.0066</td>
<td>0.0010</td>
<td>0.0371</td>
<td>0.0037</td>
<td><strong>0.0025</strong></td>
</tr>
<tr>
<td>$y_0$</td>
<td>-0.0095</td>
<td>0.0036</td>
<td>-0.0053</td>
<td>0.0488</td>
<td>0.0063</td>
<td><strong>0.0019</strong></td>
</tr>
</tbody>
</table>

Table 6.12: Model 2 photo coordinate residuals for left and right images from (a) SIFT matching, (b) DT matching, (c) manual matching:

<table>
<thead>
<tr>
<th>point</th>
<th>x left res.</th>
<th>y left res.</th>
<th>x right res.</th>
<th>y right res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS [mm]</td>
<td>0.0018</td>
<td>0.0003</td>
<td>0.0018</td>
<td>0.0003</td>
</tr>
<tr>
<td>(a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>point</th>
<th>x left res.</th>
<th>y left res.</th>
<th>x right res.</th>
<th>y right res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS [mm]</td>
<td>0.0021</td>
<td>0.0002</td>
<td>0.0021</td>
<td>0.0002</td>
</tr>
<tr>
<td>(b)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>point</th>
<th>x left res.</th>
<th>y left res.</th>
<th>x right res.</th>
<th>y right res.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS [mm]</td>
<td>0.0000</td>
<td>0.0193</td>
<td>0.0000</td>
<td>0.0193</td>
</tr>
<tr>
<td>(c)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Geometry-distribution of tie points, although they were distributed over the image, like for model 1.

2. Measuring accuracy—the points are not accurate enough. Even after several attempts, by removing points and repeating the entire measuring process, the results did not change much.
Table 6.13: Estimated relative orientation parameters and their standard deviations for manual, SIFT based and proposed approaches for OSU Model. The unit of reported values are in millimeters. Boldface indicates best results with least deviation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Manual</th>
<th>SIFT</th>
<th>Proposed</th>
<th>$\sigma_{\text{manual}}$</th>
<th>$\sigma_{\text{sift}}$</th>
<th>$\sigma_{\text{proposed}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$</td>
<td>0.3017</td>
<td>0.3328</td>
<td>0.3225</td>
<td>7.6766</td>
<td>1.0543</td>
<td>2.8364</td>
</tr>
<tr>
<td>$\phi$</td>
<td>-0.0725</td>
<td>-0.0499</td>
<td>-0.0589</td>
<td>7.3783</td>
<td>8.9327</td>
<td>6.5595</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>0.6588</td>
<td>0.6622</td>
<td>0.7651</td>
<td>4.2844</td>
<td>1.3718</td>
<td>2.9048</td>
</tr>
<tr>
<td>$X_c$</td>
<td>-0.4502</td>
<td>-0.3978</td>
<td>-0.4653</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$Y_c$</td>
<td>84.5694</td>
<td>85.1587</td>
<td>84.6832</td>
<td>0.7637</td>
<td>0.8934</td>
<td>0.6445</td>
</tr>
<tr>
<td>$Z_c$</td>
<td>154.391</td>
<td>152.718</td>
<td>153.274</td>
<td>0.7171</td>
<td>0.2763</td>
<td>0.5092</td>
</tr>
<tr>
<td>$f$</td>
<td>154.587</td>
<td>153.211</td>
<td>151.003</td>
<td>0.1854</td>
<td>0.2297</td>
<td>0.4242</td>
</tr>
<tr>
<td>$x_0$</td>
<td>-0.015</td>
<td>-0.0016</td>
<td>-0.0010</td>
<td>0.1093</td>
<td>0.0487</td>
<td>0.0728</td>
</tr>
<tr>
<td>$y_0$</td>
<td>0.012</td>
<td>0.0016</td>
<td>-0.0053</td>
<td>0.0131</td>
<td>0.0106</td>
<td>0.0110</td>
</tr>
</tbody>
</table>

In addition, for both models, DT returned a similar $\sigma_0$, as opposed to the manual or SIFT method, where this measure was inconsistent.

6.2.4 Discussion

The advantage of this method over the manual method is the consistency of the IOPs. In every instance but one in the California data, even in bundle adjustment, this method gave the $y$ coordinate of the PP a negative value, and the manual method agreed with this. For the $x$ coordinate of the PP, the proposed method gave a positive value in every instance. The manual method sometimes gave positive and sometimes gave negative values. The California camera focal length gave a more varied response as the proposed method gave conflicting values, sometimes the focal length was over 14 mm sometimes under, depending on which dataset was used; however, this is true for the manual method as well. On the other hand, the proposed method gave always consistent values for each dataset, except for one instance from image 427 when the
GIS data was fused with a generic elevation of 280 m, and one instance from bundle adjustment when the generic elevation was 275 m.

The visual results of draping the images over the terrain using the adjusted camera parameter data highlighted the differences in each method. The lidar data was reconstructed using TIN, or Triangular Irregular Network. While numerical results advise of the accuracy and precision, visualisation is a powerful technique for distinguishing relevant from non-relevant information and for locating information of interest easily and efficiently. The raw GPS/IMU values showed an offset between the images and the lidar data: the building walls from the images “slipped” onto the streets. The vegetation sometimes received roof data from the images. Using lidar only data showed a marked improvement concerning the buildings, as the building walls from the images generally fell into correct location on the lidar data. Due to the difficulty in extracting edges from lidar data, the draping is still not correct: vegetation shows up as roofs, and the walls are still noisy. Also, since only buildings were used in matching, areas between the buildings do not line up correctly between the image/lidar data. Manually selecting points also created discrepancies, as it is hard to select GCPs from lidar data. GIS fused with lidar data had similar problems as previously listed, since trees posing as roofs created a blunder that entered into the adjustment. The buildings with trees next to them had roof colors on trees, despite the tree filtering. GIS fused with a generic elevation of 270 m gave the “smoothest” walls for image 426. Since there was no noise in the elevation, and the roofs are flat, this elevation was just about the elevation of the roofs for several of the buildings. The outlines are mostly crisp. The problem in this image is that there is no elevation control on street level, so the cars near the building “melt” into the wall at times.
Interestingly, only image 426 works well with this elevation, as the elevation increases, so do the image/lidar registration errors. The noisy lidar/GIS data did not produce a much inferior product with respect to the other methods.

The error parameters were all quite good, comparable to manual matching or dense matching as presented in [93]. Since the lidar points were used as actual GCPs, the error was minimal due to the sheer number or points used, many more than in a conventional adjustment, where a handful of GCPs are used for each image.

For Model 1 manual matching, 32 evenly distributed points were selected and matched by an operator. In the SIFT-based approach the number of automatically extracted and matched points are 784. In the proposed approach, 3046 edge points are used. These points may not have exact matches in the reference image.

In Tables 6.9, 6.11, 6.13 the estimated parameters and respective standard deviations for each estimation are listed. The least deviation is observed in the proposed approach. In addition, standard deviations for the final residuals for the manual approach is 0.5746, for SIFT is 0.1253 and finally for the proposed approach is 0.0030, which is the least among the three methods.

The mean square error computed for all three methods after the adjustment is tabulated in Table 6.14. It is computed in millimeters in the image space. The manual method is the best in the $x$ direction, while the proposed method is the best in the $y$ direction.

Table 6.14: Mean square error, $\epsilon$, computed for $x$ and $y$ coordinates in millimeters in the image space.

<table>
<thead>
<tr>
<th></th>
<th>Manual</th>
<th>SIFT</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon_x$</td>
<td>0.0001</td>
<td>0.0018</td>
<td>0.0021</td>
</tr>
<tr>
<td>$\epsilon_y$</td>
<td>0.0193</td>
<td>0.0003</td>
<td>0.0002</td>
</tr>
</tbody>
</table>
However, note the $y$ mean square error for the manual approach is magnitudes higher than the proposed method.

The results reported in Tables 6.9, 6.11, 6.13 and 6.14 empirically supports the statements in [62] and [146] in that the quantity compensates for quality, thus returning a similar or better matching result. While not performed explicitly, the proposed approach uses, hence matches, considerably more points in less time on the overlapping part of the images. Overall, the parameters estimated using the manual and SIFT-based approaches, especially considering the scale of the image, is unreliable due to high standard error. This may primarily be due to:

1. **Geometry:** distribution of tie points, although they were distributed over the image.

2. **Measuring accuracy:** point locations are not accurate. For the manual approach, even after multiple attempts, such as by removing and adding points and repeating the entire measuring process, the accuracy of the parameters did not change.
Figure 6.1: High elevation variability filtered lidar. (a) White areas show constant slopes, black areas show variable slopes for an area. The black areas match vegetation and cars in the parking lot: non buildings. (b) Lidar data. Different elevations are in different colors. (c) Overlaid on lidar data.
Figure 6.2: (a) Filtering of linked edges by applying a Hough transform and 80 pixel length threshold (see text for details). (b) Lidar edges projected to image space using initial GPS/IMU values for image 425.
Figure 6.3: (a) Input image for Hough line transform after edge detection. (b) Only straight lines are of interest.

Figure 6.4: Process flow diagram of the proposed method for RO
Figure 6.5: Subset of conjugate keypoints detected and match using the scale invariant feature transform (SIFT). The ends of the green lines represent conjugate points. (a) California images 425, left, and 426, right (b) OSU images 40-01, left, 39-014, right.
Figure 6.6: (a) Gaussian noise added to GIS data (b) original building. This is the building on the right, in the middle of fig.(5.9)
Figure 6.7: Comparison of lidar points projected onto image 425 before and after adjustment of lidar data. Green pixels represent the lidar points, red outline used to enhance visibility. Local area is zoomed in to show the improvement. Projection using (a) initial camera parameters, (b) after coarse step, and (c) after refined adjustment step.
Figure 6.8: Coarse to fine parameter estimation step by minimizing (6.11) applied to GIS ground data. Local area is zoomed in to show the improvement. Projection using (a) initial camera parameters, (b) after coarse step, and (c) after refined adjustment step.
Figure 6.9: Draping comparison of various methods for Image 425
Figure 6.10: Bundle adjustment of all images overlaid on GIS.
Figure 6.11: California data relative orientation. Projection of the edges in target image and the reference image using (a) initial parameters, (b) final precise parameters. Note the increased overlap between projected edges and the reference image edges.
Chapter 7

SUMMARY AND FUTURE WORK

This research provides a novel approach to finding orientation parameters for registration of datasets. Compared to many other methods in the literature, the proposed method is resilient to viewpoint difference and occlusions, and it is feature independent. Additionally, the method resolves projective object transformations that occur due to changed viewpoint effects. The proposed approach exploits the collinearity equation via the distance function. Instead of trying to explicitly find conjugate objects, the distance function guides generic points on generic objects. There are no matching ambiguities, as there is no explicit matching. While this experiment was subject to monitoring and used some heuristic values to filter edge lengths, and was performed on a flat area, it has the potential to be automated with better results. The results have a standard error that are comparable to both the manual exterior and relative orientation. Lens distortion and other parameters may also be included in the adjustment in the future, since the adjustment produced worse results at the edge of the image. Instead of point matching, linear features are utilized for matching, and this allows data from different modalities for mosaic generation and registration/georeferencing. The proposed method implements many of the points
that were emphasized by [69], it is flexible with respect to camera type, control information, is autonomous, and comparable in speed to manual orientation, and uses more points. The accuracy of the convergence is also a reliable assessment of the success/failure of the computation. The results, based on the cost function and error distribution, is the same for LSE and MLE; however, using different functions and distributions, than those described in section(4.3.1), may result in different parameter values, as lidar errors in non-flat areas may not follow a normal distribution, or the extraction of buildings from lidar data will contain outliers that need to be modeled with a non-normal distribution. The curbs in the GIS dataset could be used as a basis for MLE probability calculation in matching to edges detected as curbs in the image. This would be much harder than matching based on just the camera parameter values used as a prior information or weight, as it is difficult to detect or threshold lidar data for such a small elevation difference, as between the road surface and curb elevation.

The registration process allows for the detection of the origin of any discrepancy between the involved datasets. The datasets in the experimental section illustrate the compatibility between lidar and photogrammetric edges. However, such compatibility will be only realized after precise calibration of both systems to guarantee the absence of systematic biases. It is of equal importance to clarify that having the two datasets registered relative to the same reference frame is a prerequisite for any further integration between the two datasets. For example, optical imagery can be draped onto the lidar data to provide a realistic 3-D textured model of the area of interest. Furthermore, registered datasets can be locally inspected for changes that took place in the object space between the data capture periods. It is imperative to mention that applying the proposed methodology is contingent on the availability of linear features,
natural or manufactured. Similar to feature-based aerial triangulation, section 2.6.2, it does not require overlapping imagery.

Creating an automatic procedure for camera calibration will enable users of digital imagery who are not experts to deliver products with high quality. As a key advantage no special landmarks or key points are necessary. It is thus a generic method with applications in object recognition, image registration and point set matching.

This is also a way to automatically compare various datasets, to ascertain what has changed in order to update maps or a GIS database. Further investigation could be registration of video data, for each frame in real time.

An alternative way to reconstruct the lidar data, instead of using a TIN, is using the distance function for a continuous surface, called level set. For comparison, a level set reconstruction of the lidar data of the same area is shown in figure(7.1). The trees in this representation look more curvy and natural, than the spike-like representation in TIN. The big trucks parked by the building also are more realistic. In a GIS, leading companies like ESRI, Mapinfo and Intergraph use a TIN as a built-in terrain modeling system. TINs are relatively easy to create and look good for smooth objects with straight or planar features, like buildings, and have other advantages, but will fail on more complex features, like a tree or person.

Future work will focus on establishing a single edge detection procedure, instead of using both EDISON and Canny. Use of priors based on geographic location will also be researched, as the proposed method would be different for urban and non-urban areas, since the detection of edges depends on the terrain and the features that are extractable from the terrain. Urban areas would have more buildings to detect, non-urban areas would have more linear features like rivers or roads that could be
used for edge detection. The use of standard deviation for each ground point or pixel should be incorporated into the method, instead of a single value for the entire area. Future work will include incorporating the lidar error budget into the adjustment, and using it for video registration.

This work focused primarily on aerial images, which image the roofs of buildings, but the side-views of buildings also need to be registered to this data to create a complete data description of the building, and the entire area, in general.
Appendix A

DRAPING IMAGERY OVER LIDAR DATA RESULTS
Figure A.1: Draping comparison of various methods for Image 425
Figure A.2: Draping comparison of various methods for Image 426
Figure A.3: Draping comparison of various methods for Image 426
Figure A.4: Draping comparison of various methods for Image 427
Figure A.5: Draping comparison of various methods for Image 427
Initial GPS/IMU values

Bundle using lidar only data

Bundle using 24 manually selected points

Bundle using GIS with lidar data

Figure A.6: Draping comparison of various methods for bundle adjustment
Figure A.7: Draping comparison of various methods for bundle adjustment
Appendix B

EXTERIOR ORIENTATION CHART

The charts below provide a visual comparison of the EOPs of the California images, 425-427.

The charts show the variable adjusted using the various methods. The methods are shown as:

PerspX: GPS/IMU data for image X

PerspXbundGIS270: bundle adjustment using GIS with 270m generic elevation data for image X

PerspXbundGIS275: bundle adjustment using GIS with 275m generic elevation data for image X

PerspXbundGIS280: bundle adjustment using GIS with 280m generic elevation data for image X

PerspXbundGISlidar2: bundle adjustment using GIS with lidar elevation data for image X

PerspXbundGISnoise: bundle adjustment using GIS with lidar elevation data with noise for image X

PerspXbundlidar: bundle adjustment using lidar data only for image X
PerspXbundman: manual bundle adjustment for image X

PerspXGIS270: adjustment using GIS with 270m generic elevation data for image X

PerspXGIS275: adjustment using GIS with 275m generic elevation data for image X

PerspXGIS280: adjustment using GIS with 280m generic elevation data for image X

PerspXGISlidar2: adjustment using GIS with lidar elevation data for image X

PerspXGISlidarnoise: adjustment using lidar data only with noise for image X

PerspXlidar: adjustment using lidar data only for image X

PerspXman: manual adjustment for image X
Figure B.1: Adjustment results 425
Figure B.2: Adjustment results 426
Bibliography


[121] ODNR Ohio Department of Natural Resources. Geographic information policy and guidelines of the ohio department of natural resources, 2008. Brochure.


213


