

Electronic Thesis and Dissertation Repository

4-17-2013 12:00 AM

Improving the Geotagging Accuracy of Street-level Images

Mehrnaz Zouqi

The University of Western Ontario

Supervisor

Dr. Jagath Samarabandu

The University of Western Ontario

Graduate Program in Electrical and Computer Engineering

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

© Mehrnaz Zouqi 2013

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Zouqi, Mehrnaz, "Improving the Geotagging Accuracy of Street-level Images" (2013). *Electronic Thesis and Dissertation Repository*. 1202.

<https://ir.lib.uwo.ca/etd/1202>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.

Improving the Geotagging Accuracy of Street-level Images

(Thesis format: Monograph)

by

Mehrnaz Zouqi

Faculty of Engineering
Department of Electrical and Computer Engineering

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

© Mehnaz Zouqi 2013

Abstract

Integrating images taken at street-level with satellite imagery is becoming increasingly valuable in the decision-making processes not only for individuals, but also in business and governmental sectors. To perform this integration, images taken at street-level need to be accurately geotagged. This geotagging information can be derived from a global positioning system (GPS). However, GPS data is prone to errors up to 15 meters, and needs to be corrected for the purpose of geo-referencing. In this thesis, an automatic method is proposed for correcting the geotag information obtained from the GPS data, based on image registration techniques. The proposed method uses an optimization technique to find local optimal solutions by matching high-level features and their relative locations. A global optimization method is then employed over all of the local solutions by applying a geometric constraint.

The main contribution of this thesis is introducing a new direction for correcting the GPS data which is more economical and more consistent compared to existing manual method. Other than high cost (labor and management), the main concern with manual correction is the low degree of consistency between different human operators. Our proposed automatic software-based method is a solution for these drawbacks.

Other contributions can be listed as (1) modified Chamfer matching (CM) cost function which improves the accuracy of standard CM for images with various misleading/disturbing edges; (2) Monte-Carlo-inspired statistical analysis which made it possible to quantify the overall performance of the proposed algorithm; (3) Novel similarity measure for applying normalized cross correlation (NCC) technique on multi-level thresholded images, which is used to compare multi-modal images more accurately as compared to standard application of NCC on raw images. (4) Casting the problem of selecting an optimal global solution among set of local minima into a problem of finding an optimal path in a graph using Dijkstra's algorithm.

We used our algorithm for correcting the geotagging information of 20 chains containing more than 7000 fisheye images and our experimental results show that the proposed algorithm can achieve an average error of 2 meters, which is acceptable for most of applications.

Keywords

Geotagging, Street-level Fisheye Image, Modified Maximization of Mutual Information Registration Method, Line Descriptor, Modified Chamfer Matching Registration Method, Dijkstra's Algorithm

Acknowledgements

First of all, I would like to thank my supervisor Dr. Samarabandu who has helped me throughout the work. I also thank Dr. Jeff Weaver, Dr. Yanbo Zhou and Jeff Hack at iLookabout Inc. who defined this project and supported it financially and scientifically.

I would also like to thank my colleagues in Computer Vision and Mobile Robotics Lab for their friendship and support.

My special thank goes to my lovely husband, Mahdi who has always supported me during the hard times.

Finally, I present my deep gratitude to my family and I would like to dedicate this thesis to my beloved parents; without their sacrifices, encouragements and love I would have never gone this far in my life.

Contents

ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	x
1 Introduction	1
1.1 Applications of street-level imaging	3
1.2 Collecting street-level images	5
1.3 Global Positioning System	6
1.4 Problem statement	7
1.5 Overview of thesis	8
2 Street-level Fisheye Images	10
2.1 Radial distortion	11
2.2 Calibration	14
2.3 Vacant pixels/ sparse problem	15
3 Related Work	17
3.1 Current solution for GPS data correction	18
3.2 Image Registration	19
3.2.1 Variation-based classification	21

3.2.2	Application-based classification	21
3.2.3	Methodology-based classification	22
3.3	Mutual Information Based Registration	23
3.3.1	Information entropy and uncertainty	24
3.3.2	Entropy of an image	25
3.3.3	Joint Entropy	25
3.3.4	Mutual Information	26
3.3.5	Our Modifications to MMI	26
3.4	Why not MMI?	27
4	Data Preparation/ Description	30
4.1	Fisheye stripe	30
4.2	Satellite stripe	32
5	Modified Chamfer Matching	36
5.1	Integration of edge orientation information	37
5.2	Integration of edge strength	38
6	Methodology	42
6.1	Image Selection	43
6.2	Registration method	44
6.2.1	Detection of Line Correspondences	44
6.2.2	Modified Chamfer Matching	46
6.2.3	Optimizing Stripe Alignment using Dijkstra’s Algorithm	47
6.3	Rejection of Weak Registration Results	48
7	Experiments and Results	52
7.1	Experiment 1	53
7.2	Experiment 2	54
8	Software Architecture	57
9	Concluding Remarks and Future Work	61
9.1	Discussion	61

9.2 Thesis Contribution	63
9.3 Future Research	64
A GPS and Its Sources of Error	66
B Graph Cuts	70
C Mean Standard Deviation Line Descriptor	72
D Chains' Information	75
E Results	95
VITA	119

List of Tables

7.1	Results: E_{CM} , E_{MCM} are the average errors from standard Chamfer matching method and our modified Chamfer matching method. E_{final} shows the final error of the algorithm after rejection of weak registration results using NCC	54
E.1	Results for chainId=45614	95
E.2	Results for chainId=45616	96
E.3	Results for chainId=55764	96
E.4	Results for chainId=55938	97
E.5	Results for chainId=55938 (continue)	98
E.6	Results for chainId=55964	99
E.7	Results for chainId=93849	100
E.8	Results for chainId=151323	100
E.9	Results for chainId=151325	101
E.10	Results for chainId=284553	102
E.11	Results for chainId=286309	103
E.12	Results for chainId=286313	104
E.13	Results for chainId=287607	104
E.14	Results for chainId=112357	105
E.15	Results for chainId=112357 (continue)	106
E.16	Results for chainId=114759	107
E.17	Results for chainId=114759 (continue)	108
E.18	Results for chainId=114761	109
E.19	Results for chainId=230253	110

E.20 Results for chainId=290663	110
E.21 Results for chainId=53190	111
E.22 Results for chainId=53456	112
E.23 Results for chainId=54285	112

List of Figures

1.1	GIS applications (Image courtesy of Esri website [1])	3
1.2	Google Street View technology	4
1.3	Street-level imaging car	6
1.4	A sample chain on a satellite image.	7
2.1	A sample street-level fisheye image	11
2.2	(a) Ideal pin-hole camera model (b) Fisheye distortion	12
2.3	Principle of equidistant projection method [26].	13
2.4	Principle of central projection method [26].	14
3.1	Main steps of the proposed MMI-based registration technique	28
3.2	Example of failure of MMI-based registration methods. (a) Part of satellite image. (b),(c) Fisheye images with road-like features. (d) Fisheye image with real road. MI score between (a) and ((b) or (c) or (d)) is the same.	29
4.1	Stitching two images can be converted to an optimization prob- lem to be solved by graph cuts. Image courtesy of V. Kwatra et al. [28]	31
4.2	Seven fisheye images (shown in the top row)were stitched to- gether seamlessly to make a fisheye stripe (shown in the bottom row).	31
4.3	Quadkeys- binary representation: Tile system of the satellite image of the whole world in different resolutions.	33
4.4	Quadkeys- letter representation.	33

4.5	Part of a sample chain and the extracted satellite stripe	35
5.1	An example of (a) a binary image and (b) its distance map. . .	37
5.2	Construction of the edge map for the satellite stripe: (a) a sample satellite stripe (b) binary image obtained using Otsu's method (c) binary image obtained using threshold = Otsu's threshold + 0.1 (d) edge map	39
5.3	(1 st row): sample fisheye images from one chain. (2 nd row): binary conversion using Otsu's method. (3 rd row): binary conversion using Otsu's threshold+offset(0.3, 0.3, 0.3, 0.15, 0.1, 0.25 respectively). (4 th row): our proposed edge map based on multi-thresholding.	40
5.4	Construction of the multi-level edge map for a sample fisheye stripe.	41
6.1	Overall view of the proposed method	43
6.2	Model image to detect the vertical features on fisheye images . .	43
6.3	Overall view of the proposed registration method	45
6.4	The process of detecting similar horizontal lines	46
6.5	Finding the optimal path between the local minima using Dijkstra's algorithm.	48
6.6	Image selection algorithm may fail, due to shadows that generate road-like features.	49
6.7	Satellite stripes and their corresponding fisheye stripes. top row: driveways are occluded. bottom row: driveways are not identifiable due to construction.	50
6.8	Evaluation of final result for a sample pair of images.	51
7.1	Some of the final results for <i>chainId</i> = 45614 (a) Fisheye stripe (b) Final satellite stripe (c) original, manually-corrected and algorithm-corrected coordinates shown in blue, red and green respectively.	55

7.2	Error distribution of the proposed registration algorithm for all the images in 20 chains	56
8.1	Functionality of the proposed algorithm	57
8.2	Overall architecture of the software for the proposed geotagging algorithm based on image registration	60
A.1	Errors of GPS signal	67
A.2	Tall buildings cause the reflection of satellites' signals (Image courtesy of Kowoma website [2])	69
B.1	A simple 2-D graph for a 3×3 image and its minimum cut.	71
C.1	Schematic figure of MSLD construction [46]	74
D.1	<i>Chain_Id</i> = 45614	75
D.2	<i>Chain_Id</i> = 45616	76
D.3	<i>Chain_Id</i> = 55764	77
D.4	<i>Chain_Id</i> = 55938	78
D.5	<i>Chain_Id</i> = 55964	79
D.6	<i>Chain_Id</i> = 93849	80
D.7	<i>Chain_Id</i> = 151323	81
D.8	<i>Chain_Id</i> = 151325	82
D.9	<i>Chain_Id</i> = 284553	83
D.10	<i>Chain_Id</i> = 286309	84
D.11	<i>Chain_Id</i> = 286313	85
D.12	<i>Chain_Id</i> = 287607	86
D.13	<i>Chain_Id</i> = 112357	87
D.14	<i>Chain_Id</i> = 114759	88
D.15	<i>Chain_Id</i> = 114761	89
D.16	<i>Chain_Id</i> = 230253	90
D.17	<i>Chain_Id</i> = 290663	91
D.18	<i>Chain_Id</i> = 53190	92
D.19	<i>Chain_Id</i> = 53456	93

D.20 <i>Chain_Id</i> = 54285	94
----------------------------------------	----

Nomenclature

2D - Two Dimensional

3D - Three Dimensional

CM - Chamfer Matching

E-government - Electronic government

FOV - Field Of View

GC - Graph Cuts

GIS - Geographic Information System

GPS - Global Positioning System

Inc. - Incorporation

km - kilometer

m - meter

m/s - meters per second

MCM - Modified Chamfer Matching

MI - Mutual Information

MMI - Maximization of Mutual Information

MSLD - Mean Standard Deviation Line Descriptor

NCC - Normalized Cross Correlation

OCM - Orientational Chamfer Matching

SA - Selective Availability

Chapter 1

Introduction

Geographic Information System (GIS) is a computer-aided system that integrates hardware, software and data to convert the information about the real world into a digital database of spatial and nonspatial features [35]. GIS allows to view, understand, interpret, and visualize data in many ways that reveal relationships, patterns, and trends in the form of maps, globes, reports, and charts [5]. GIS technology can be integrated into any enterprise information system framework. For instance, GIS can optimize the use of city data, such as population, ethnocultural, education, immigration, etc. in solving difficult business problems. Websites that provide neighbourhood statistics in where you live (such as ward profile summary¹) are examples of using GIS.

Everyday, urban planners use GIS technology to research, develop, implement and monitor the progress of their plans [1]. E-government is also using GIS to provide services to businesses, citizens and various levels of government [1].

Government-to-business applications are services such as, economic development, land use, licensing, permitting, etc.

Government-to-citizen applications are services such as, trash pickup, online payment of fees, providing feedback, etc.

Government-to-government applications provide better communication between different agencies and departments of the government to avoid duplication of effort and allows them to work together more efficiently. For example GIS

¹<http://app.toronto.ca/wards/jsp/wards.jsp>

can make property tax assessment simpler by bringing spatial element to the information that IT department provides, and prioritizes areas based on the tax revenues collected or due [16].

Benefits of using GIS in local government include the following [1]:

- Increase efficiency
- Generate revenue
- Improve accuracy
- Automate tasks
- Increase access to government
- Promote greater collaboration among public agencies
- Save time and money
- Provide decision support
- Manage resources
- Enhance public participation

Overall, GIS facilitate the decision-making process (look at figure 1.1). However, rapidly changing socio-economic and environmental conditions have further complicated this process. In order to reduce these complications, urban planners have started to integrate spatial information in planning tools through high resolution street-level imagery [31].

Street-level imaging has its origin in Street View technology introduced by Google in 2007 to provide 360 degree panoramic views of the selected locations on the map. Since then, this technology has been used and expanded by other companies (e.g. *Streetside* from Microsoft and *StreetScape* from iLookabout Inc.). As figure 1.2 shows, within Street View, users can follow the direction of the roads to move virtually up and down the streets. It is also possible to zoom in on street signs to for example read signs indicating when parking



Figure 1.1: GIS applications (Image courtesy of Esri website [1])

is allowed. Therefore, street-level imaging represents a new geographic information systems (GIS), providing details that are not possible with satellite or aerial photography.

1.1 Applications of street-level imaging

Street-level imaging has applications in economic development, real estate, city planning, tourism, facilities management, and construction management. Geo-spatial companies are the supplier of high-resolution 360-degree geo-coded street-level imaging. Using a proprietary system, their customers have access to geo-coded panoramic image databases through a license agreement, which allows for sharing images on a web server in Intranet or Internet environments. They also provide web hosting services for customers who want a turn-key solution that includes image collection, application deployment, and web access. Their products support activities such as [4]:

- Asset Inventory and Management
- Permit and Code Enforcement

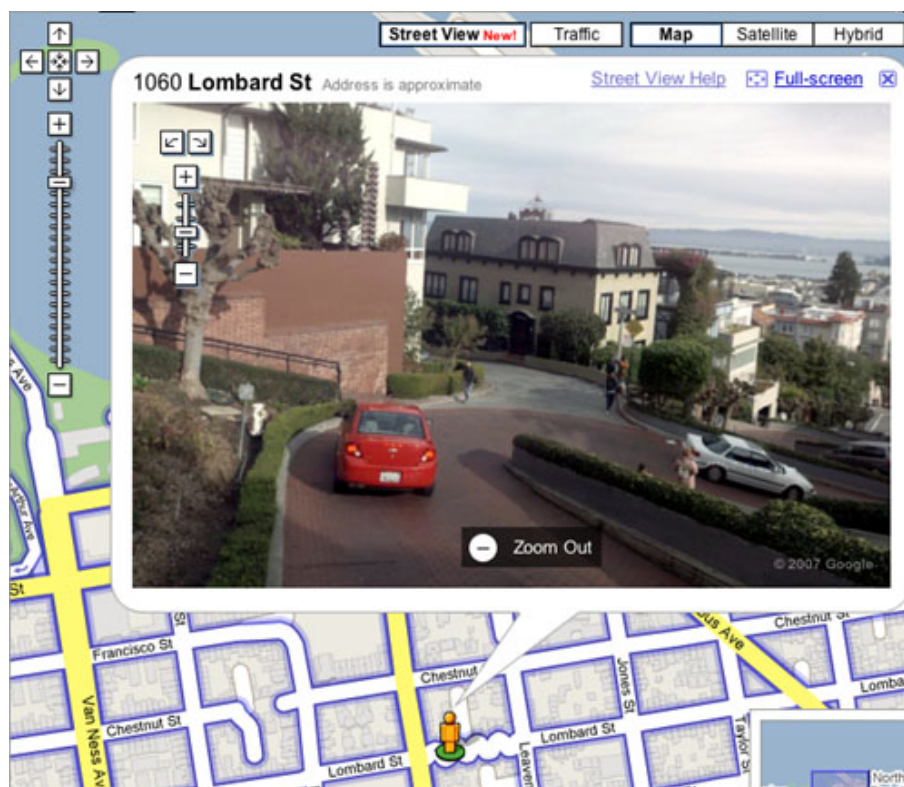


Figure 1.2: Google Street View technology (Image courtesy of Digitaltrend website²)

- Public Works Maintenance
- Engineering and Inspection
- Public Safety/Fire/Police Activity Planning and Staff Training
- City Planning and Economic Development
- Facility Management

For example Public Works departments can check on the location of traffic signs, buildings, storm drains, and other structures in support of asset inventory and assessment activities. They can even examine the location details

²<http://www.digitaltrends.com/computing/google-maps-adds-street-level-images/>

such as pavement conditions. As another example, city planners have the capability to see actual street and sidewalk conditions related to environmental impact and zoning reviews. This has proven to be a time and cost saving asset in an annexation project by reviewing digital photos of the area without driving to the project site. As the street-level images are geo-coded, they are easily integrated with aerial photography and other geospatial database sources. For example they can be integrated with public safety geospatial databases to support crime or fire suppression response strategies.

Mortgage providers are another customer of geo-spatial imaging companies. High resolution images of the buildings help lenders decrease loan processing time and quickly identify external issues with properties. Within seconds, they will know the condition of the property's exterior and whether it conforms to the rest of the neighborhood. With such insight, lenders are better prepared to make decisions³.

1.2 Collecting street-level images

Street-level images are taken by a vehicle with a number of cameras mounted on its roof and automatically capture images at a certain distance. The vehicle (Figure 1.3) drives around the streets of cities and collects images. Images are collected and stored as a *chain*. A chain of images is a series of images taken along one street block. For example, Figure 1.4 shows a sample chain on a satellite image.

To show the street-level images on the right locations of the satellite image (or map), the images need to be geotagged. Geotagging is the process of adding geographical identification data (spatial location) to the images⁵. Spatial location can be obtained from an on-board Global Positioning System (GPS) receiver.

³<http://www.businesswire.com/news/>

⁴<http://www.telegraph.co.uk/technology/google/7711297/Google-rebuffs-privacy-concerns.html>

⁵<http://en.wikipedia.org/wiki/Georeference>



Figure 1.3: Street-level imaging car (Image courtesy of Telegraph website⁴)

1.3 Global Positioning System

The basic components of the GPS are satellites and receiver. Each satellite continually transmits messages that include the time the message was transmitted and satellite position at the time of message transmission. A GPS receiver compares the time when the signal was sent by the satellite with the time the signal was received. Using this time difference and the speed of light, the distance between receiver and satellite can be calculated. If data from other satellites are taken into account, the GPS receiver can calculate its position by trilateration. In geometry, trilateration is the process of determining absolute or relative locations of points by measurement of distances, using the geometry of circles, spheres or triangles⁶. This means that at least three satellites are required to determine the position of the GPS receiver on the earth. More details about the GPS system can be found in Appendix A. A standard GPS receiver for civil use provides an accuracy down to ± 15 meters. Some

⁶<http://en.wikipedia.org/wiki/Trilateration>



Figure 1.4: A sample chain on a satellite image.

of the sources of errors are multipath effects, atmospheric effects and clock errors. Appendix A explains the sources of errors for GPS in more details.

1.4 Problem statement

Accurate position estimate of street-level images is essential for associating them with the map and for enabling an intuitive navigation experience. Various businesses are relying on the accurate geotagged street-level images. The accuracy that these businesses require is higher than what GPS provides. For example, the maximum tolerable error for iLookabout Inc.⁷ is 2.5 meters.

This thesis is an attempt to develop a software-based method to automate the process of accurate geotagging of street-level fisheye images by aligning them with an accurate reference of satellite imagery, using novel image registration techniques.

⁷<http://www.ilookabout.com/>

The main contributions of this thesis is that we propose a software-based method to increase the accuracy of geotag data for the street-level fisheye images. As it will be explained, original geotag data that is obtained from GPS can be corrected by odometry data. However, accuracy that odometry provides is not enough for some applications and manual correction is the current solution to achieve the required accuracy. Our proposed method is an image-based method to automate the manual correction process. Therefore, the method is more economical and the consistency is higher than the manual correction. We improve the accuracy of geotag information by registering the street-level images into the satellite image. Our new image registration framework is appropriate for other multi-modal images with occlusion and various background clutter.

Other contributions that will be explained throughout the thesis are as follows:

- Our proposed Modified Chamfer matching method introduces a new set of weights to the objective function of the standard CM score to undermine the weak false features. (and to preserve the solution when the false features are stronger than the correct features.)
- Our proposed Monte-carlo-inspired statistical analysis is a novel method to quantify the overall performance of an image registration algorithm.
- Applying NCC on multi-level thresholded images instead of raw intensities is a new way of evaluation with better performance for our images.

1.5 Overview of thesis

In chapter 2, fisheye concept in imaging will be explained. We will see that the fisheye distortion makes the images more challenging for registration purposes, and needs to be taken care of before developing any image registration method.

Chapter 3 contains the existing methods for geotagging which are mainly based on data fusion between GPS data and information coming from different sensors installed on the vehicle collecting the images. Then a review of the related

literature on image registration techniques is given, as our proposed geotagging method is based on image registration. Particularly, MMI-based methods will be explained in more details at the end of that chapter.

In chapter 4 the preparation of satellite and fisheye images is given, including extraction of satellite image of a specific chain from google maps and stitching small top-view of street-level fisheye images together to make a stripe.

The core of our proposed algorithm, which is modified Chamfer matching method, will be explained in chapter 5.

Our proposed algorithm will be described in chapter 6, and the experimental results can be found in chapter 7. In chapter 8, the details of the implementation of the proposed algorithm is given. Finally, in chapter 9, we conclude the thesis with discussion, contribution and possible directions for the future work.

Chapter 2

Street-level Fisheye Images

In computer vision literature, the term “conventional camera” usually means pin-hole or planar-perspective cameras. A conventional camera has a very limited field of view (FOV). However, in many computer vision applications, including robot navigation, 3D reconstruction, image-based rendering, and single view metrology, a camera with a quite large FOV is preferable. Therefore, “omnidirectional cameras”, such as a fisheye camera, are employed, which have a larger field of view (i.e. more than or equal to 180 degrees). Figure 2.1 shows an image taken with a fisheye camera. Although fisheye cameras have the advantage of capturing more than 180 degree field of view, but images taken with these cameras have severe forms of distortion, such as uneven illumination, inaccurate estimation of the centre of distortion and the most evident one: *radial distortion*, which causes the straight objects (lines) appear bend in the fisheye image [25].

As radial distortion can cause problems for image processing techniques, distortion correction is an important pre-processing task for computer vision applications. In this chapter, we briefly explain how the correction can be done by modeling the distortion through calibration of the camera, and then using the inverse of the model to transform the pixels to their ideal locations in pin-hole camera model.



Figure 2.1: A sample street-level fisheye image

2.1 Radial distortion

Radial lens distortion causes points in the fisheye image to be displaced in a non-linear manner from their ideal position in the pin-hole camera model, along a radial axis from the centre of distortion in the image plane [25]. The visual effects of this displacement is shown in Figure 2.2. The aim of distortion correction is to transform the distorted fisheye image to the ideal pin-hole perspective view [25].

Hughes et al., in their review paper [25], have categorized the standard fisheye radial distortion models into two categories:

- Polynomial models
- Non-polynomial models

In the first category, distortion is modelled as a mapping from ideal pin-hole image to fisheye image plane, via a radial dependent polynomial. If the projection of a 3-D object point on the pin-hole image plane is represented as an undistorted point, $p_u = (x_u, y_u)$, and its corresponding distorted point on the fisheye image plane is represented as $p_d = (x_d, y_d)$, then the radial distances for ideal and distorted points are calculated as $r_u = \sqrt{x_u^2 + y_u^2}$ and

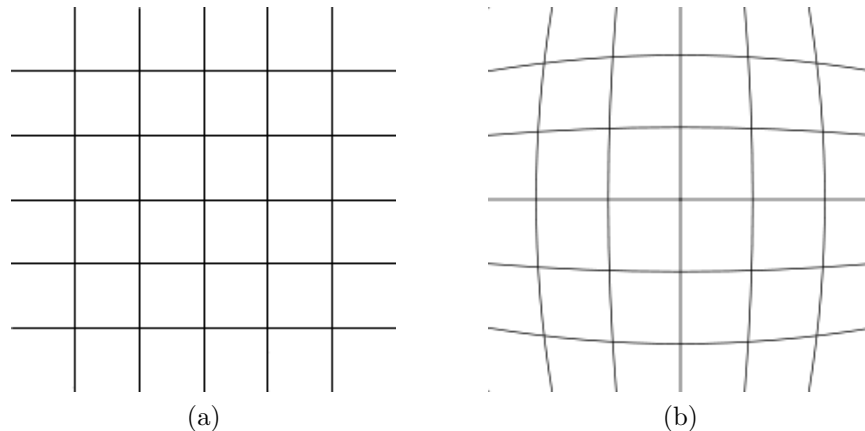


Figure 2.2: (a) Ideal pin-hole camera model (b) Fisheye distortion

$r_d = \sqrt{x_d^2 + y_d^2}$ [36]. Now, a polynomial that can be used to model the radial distortion of a fisheye lens is as follows [25]:

$$r_d = \sum_{i=1}^{\infty} a_i r_u^i \quad (2.1)$$

where a_i are scalar coefficients. Although, there is no general analytical method to invert 2.1, approximation can be used. For example, by neglecting the coefficients beyond the fifth order, Mallon and Whelan [36] approximated the inverse of an odd polynomial as follows:

$$r_u = r_d - r_d \left(\frac{a_1 r_d^2 + a_2 r_d^4 + a_1^2 r_d^4 + a_2^2 r_d^8 + 2a_1 a_2 r_d^6}{1 + 4a_1 r_d^2 + 6a_2 r_d^4} \right) \quad (2.2)$$

Compared to polynomial models, fisheye distortion models that are not based on a polynomial approximation of the fisheye lens distortion curve are analytically invertible in general [25]. One of these models that is used frequently for radial distortion is “perspective model”. This model is based on the “equidistant projection rule” which says that the height of length of an object in a fisheye image (r_d) is equivalent to the product of the angle from optical axis $\theta(\text{rad})$ and focal length $f(\text{mm})$ [26]:

$$r_d = f \times \theta \quad (2.3)$$

For a standard pin-hole camera, the projection rule is formulated as follows:

$$r_u = f \times \tan\theta \quad (2.4)$$

where, r_u is the height of length of an object in an image, f is the focal length and θ is the angle from optical axis. Figures 2.3, 2.4 show these two equations graphically.

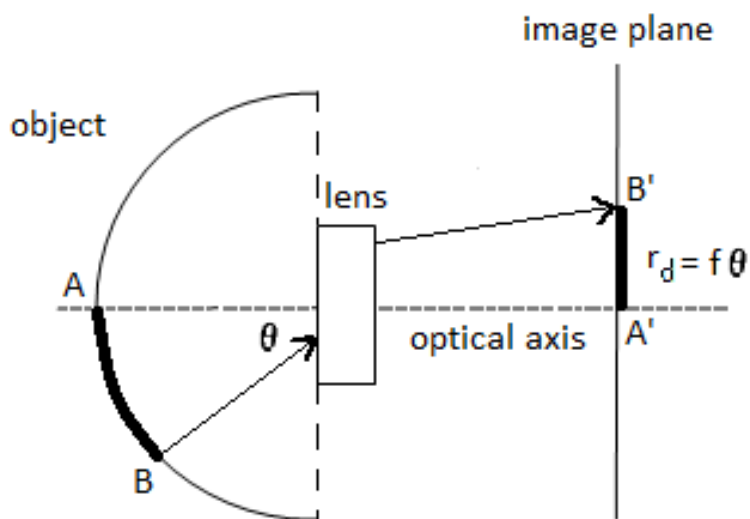


Figure 2.3: Principle of equidistant projection method [26].

From equations 2.3 and 2.4, the perspective model and its inverse will be derived as follow:

$$r_d = f \times \arctan\left(\frac{r_u}{f}\right) \quad (2.5)$$

$$r_u = f \times \tan\left(\frac{r_d}{f}\right) \quad (2.6)$$

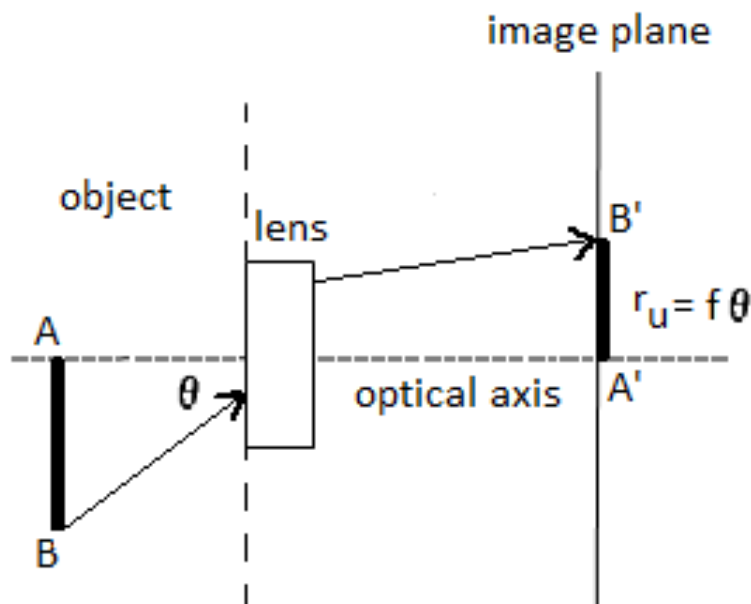


Figure 2.4: Principle of central projection method [26].

2.2 Calibration

In general, the correction of radial distortion involves a calibration process to determine the parameters of one of the fisheye models, described in the previous section. Then, the distortion can be corrected by inverting the model and transforming each pixel in the image according to that model [25]. *Photogrammetric* calibration is a standard procedure that uses features such as corners, dots, lines and circles to calibrate the camera.

For example, Basu and Licardie [8] measured the amount of distortion caused by a fisheye lens by using a set of fisheye images taken from a grid. They first determined the center of distortion by finding the intersection of lines with minimum curvature in the orthogonal direction. To obtain a better estimate of the center of distortion, they averaged the several estimates from 10 different grid images. Then, using the same grid images, they obtained a measure of distortion in four different directions (up, down, left and right) and averaged those values to obtain a more reliable estimate. The measure of distortion is the radial distance (in pixels) between a pixel of the grid and the center of

distortion. It is determined by comparing the expected position of such a pixel in pin-hole camera image and its measured position in the fisheye image [8].

2.3 Vacant pixels/ sparse problem

The circular shape of a fisheye image is an evidence that if you move up the image the distance from the y-axis to the edge of the image decreases in pixels [37]. Therefore, any radial distortion correction method necessitates stretching on the corners to make a rectilinear image (look at Figure 2.5). Due to this “stretching” effect during the distortion correction, resultant undistorted image will contain many vacant pixels that will not have been mapped during the correction process. Interpolation methods can be used to solve this problem [25]. Alternatively, *back-mapping* may be used as well. Back-mapping takes the location of the pixel in undistorted image and calculates its location in distorted image using the forward transform model. This way, every pixel in the undistorted image will be assigned a value [25].

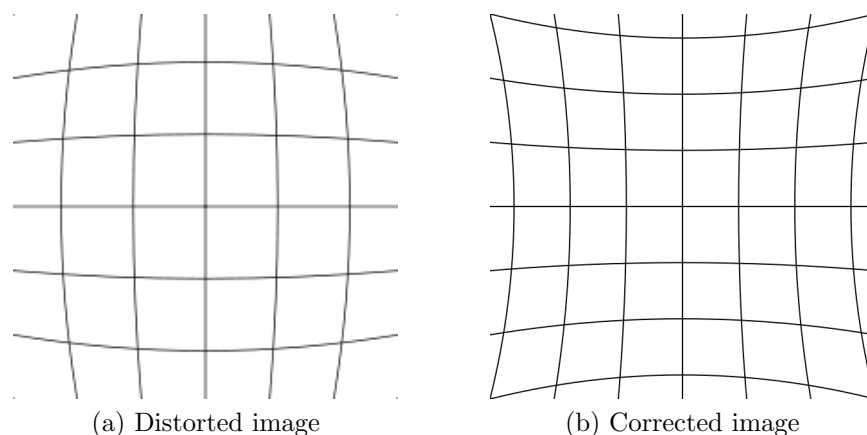


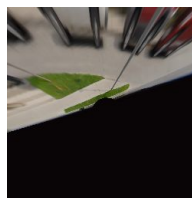
Figure 2.5: Simulation of the correction of radial distortion

Producing a wide-angle image free of distortion is challenging because there is no global projection which is free of all types of perceptual distortion [15]. Zorin and Barr [49] proved that no global projection can both keep straight

lines straight and preserve the shapes of objects. Different projections are useful for different applications. In our case, as we are interested in straight lines for our image registration method, perspective projection seems a reasonable choice, as it preserves the straightness of the lines in the scene. However, objects located near the periphery of wide-angle perspective images can appear unnaturally stretched and distorted [15]. To solve this problem, we worked with a small region of the “geometrically corrected” fisheye images (called top-view) to avoid this highly distorted peripheral region. As shown in Figure 2.6(b), this top-view is more similar to a satellite image and makes the registration process easier.



(a) Street-level fisheye image



(b) Top-view

Figure 2.6: (a) An example of a street-level fisheye image and (b) a section of its geometrically corrected image.

Chapter 3

Related Work

Currently, the solution of geospatial companies to fix the limited accuracy of the GPS data is a combination of sensor-based and manual correction. As it will be explained in section 3.1, they first fuse the GPS data with data obtained from different sensors on the vehicle (odometry) to achieve a better accuracy. If more accuracy is needed, they have to obtain it by manually aligning the street-level images with the satellite image. There are a number of drawbacks related to manual correction, including high cost (of labor and management), time required, low degree of consistency, and limited control of quality. These drawbacks are driving businesses in the field of geo-spatial imaging towards increased automation of the correcting process of GPS data for the street-level images they acquire.

Our proposed algorithm in this thesis is an *automatic software-based* method to geotagged street-level fisheye images. The core of our proposed algorithm is based on registering the street-level images into satellite image. Therefore, in section 3.2 we will review existing image registration algorithms.

Section 3.3 includes the details about one of very successful multi-modal image registration methods (which is maximization of mutual information), and hence our first choice to register street-level images into satellite image. We modified the original MMI method to increase reliability and efficiency. Later on, after working with a large set of images, we realized that there are deficiencies with all the intensity-based algorithms and they can not handle all

type of issues our images possess. The reasons that we did not develop our final algorithm based on MMI will be explained in section 3.2. In spite of the fact that our final solution is not based on MMI, but the modifications that we have made to standard MMI are valuable, and can be used by other researchers, wherever the data is appropriate for an intensity-based registration method.

3.1 Current solution for GPS data correction

Currently, odometry is the main solution to increase the accuracy of GPS data. Odometry is the use of data from moving sensors to estimate change in position over time, relative to a starting location. This method is sensitive to errors due to the integration of velocity measurements over time. Rapid and accurate data collection, equipment calibration, and processing are required for odometry to be useful¹. In most cases odometry does not provide the required accuracy for the GPS data, and manual correction is considered to achieve the desired accuracy.

For instance, Google Inc. uses the GPS, wheel encoder, and inertial navigation sensor to geotag its Street View images [6]. Then, as part of the Street View processing pipeline, they use their open-source algorithm² to achieve a smoother and locally accurate solution for the pose trajectory. Once the pose based on sensor fusion is computed, they align the images to the map of the road network. After all these processings, the locations are not totally error-free yet. Hence, Google Inc. has recently opened Street View to user contributions. Users can correct the location of businesses and points of interest by dragging markers in Street View and automatically snapping it to facades [6]. With this ability, users can build map data with greater accuracy. iLookabout Inc. employs almost similar strategy as Google. They have developed a proprietary algorithm that can enhance the latitude and longitude provided by the GPS to some extent using the distance information. Distance

¹<http://en.wikipedia.org/wiki/Odometry>

²<http://code.google.com/p/gpo/wiki/GPO>

sensors are mounted on two wheels of the vehicle that regularly send pulses. These pulses are recorded along with the images to calculate the distance that the car has passed by counting the number of pulses. Using these signals increases the accuracy of the GPS data along the chain to some extent. However, there is still a need for manual correction (specially perpendicular to the chain) to fix the remaining errors which can be up to 6 meters. To achieve a better accuracy, manual correction will be done next. For manual correction, human operators find the location of the fisheye images on the satellite image by matching common features such as corners of grass, driveways, etc. Images that do not have any specific features are located based on their distance to those that can be matched (fixed points). Basically, they only fix the locations of a few images within a chain and propagate the error among the rest of the images.

3.2 Image Registration

The need to register images has arisen in many practical problems in diverse fields. Registration is often necessary for integrating information, finding changes, inferring 3D information, object recognition, etc. The aim of image registration is matching two (or more) images of the same scene taken at different times, at different resolution, from different view points, and/or by different sensors [48]. Mathematically, image registration is a mapping between two images both spatially and with respect to intensity [13]. If the 2D images are denoted by I_1 and I_2 , then the registration means finding C and T in the following equation:

$$I_2(x, y) = C(I_1(T(x, y))) \quad (3.1)$$

where T is a 2D spatial-coordinate transformation and C is a 1D intensity transformation. Intensity transformation is not always necessary, and often a simple lookup table determined by sensor calibration techniques is sufficient [9]. Therefore, finding the parameters of the spatial or geometric trans-

formation T is generally the key part of the registration problem to remove the geometric variations between images. Note that geometric variations can be divided into two groups: Those we want to correct for and those we don't [13]. For example, variation due to viewpoint changes should be removed, but variations due to changes in the scene that are of interest should be preserved (such as monitoring tumor growth). When designing a registration algorithm, it is important to identify the variations we want to correct for.

Over the years, a broad range of image registration techniques has been developed for various types of data and applications, resulting in a large body of literature [13]. This broad spectrum of methodologies makes it difficult to classify and compare techniques since each technique is often designed for specific applications and not necessarily for specific types of problems or data. However, most registration techniques involve searching over the space of transformations of a certain type to find the optimal transformation for a particular problem. Determining the best spatial transformation can be broken down into the selection of the appropriate choice/technique for the following four components [13]:

- feature space: which information in the image is used for matching (e.g. pixel intensities, edges, feature points)
- search space: which kind of transformation(s) are required to align the images (e.g. translation, rigid, similarity, affine, projective)
- search strategy: which optimization technique is used to find the parameters of the optimal transformation (e.g. exhaustive, simplex method, dynamic programming)
- similarity metric: how the search strategy measures the goodness of each result (e.g. correlation score, mutual information score, Chamfer matching score)

For each specific problem, a combination of choices for these four components can be selected to develop the most suitable registration technique.

3.2.1 Variation-based classification

Brown [13] has organized the existing registration techniques by establishing a relationship between the variations (or distortions) in the images and the choices for the four components of image registration. So, the types of variations present in the images will determine the selection of the appropriate technique for each of these components.

Different types of variations include the following [13]:

- sensor noise
- spatial distortions due to change of viewpoint
- perspective changes from sensor view-point or platform perturbations
- object changes such as movements, deformations, growths, or other scene changes
- lighting and atmospheric changes including shadows and cloud coverage
- different sensors

Some of these variations like the spatial distortions can be satisfactorily removed by a correct transformation. However, some of them like object movements or shadows can not be modeled easily, and make registration more difficult. We have to limit the influence of these distortions by choosing the appropriate feature space, similarity measure, and search strategy.

All of the above variations exist in the problem of registering street-level fisheye images into satellite image, which shows that the level of difficulty/complexity for this problem is high.

3.2.2 Application-based classification

Another classification method for image registration literature can be based on application; the application itself specifies the choices of algorithms. In this

regard, we can have the following subdivisions which are partially based on the work of Elsen et al. [20]:

- **Modality:** *Mono-modal* refers to the images that are obtained from the same sensor type and there are no major differences between the intensity ranges of the same structures in the images. On the other hand, in *Multi-modal* imaging, these ranges can differ drastically, usually due to different sensor types that has been used.
- **Dimensionality:** This refers to the number of dimensions of the images: 2D images, 3D volumes. Also, when working with videos, time can be considered as an extra dimension.
- **Speed:** Regarding speed, applications can be divided into *offline* and *online*. *Offline* refers to applications where time is not an important constraint, contrary to *online*.

3.2.3 Methodology-based classification

From the methodology point of view, registration algorithms can be classified based on the following criteria:

- **content-based vs. feature-based:** The information employed for registration can be categorized in two groups of content-based and feature-based [23]. In feature-based approaches, alignment is done based on landmarks such as sets of points, lines, surfaces, etc. These landmarks can have physical meaning (e.g. driveways, road intersections, etc.), or they can be geometrical features (e.g. corners, points of high curvatures, etc.). On the other hand, content-based approaches compute an alignment based on pixel intensity information such as intensity values [17, 27, 44], gradient vectors [21], wavelet coefficients [30, 32, 43], etc. Feature-based approaches are usually faster than content-based ones. Because, they just work with a sparse set of features compared to the original image content. However, the performance of feature-based

methods heavily depends on the accuracy and reliability of the feature correspondences.

- **Transformation:** Registration techniques can also be categorized regarding their search space (transformation type). General transformations are rigid, affine, projective, perspective, and global polynomial. Transformations used to align two images may be global or local. A global transformation is given by a single equation which maps the entire image. Local transformations map the image differently depending on the spatial location³.

In this thesis, only 2D global rigid transformation is considered which is composed of a scaling, a translation, and a rotation. Hence, it has four parameters, t_x , t_y , s , θ , which map a point (x_1, y_1) of the first image to a point (x_2, y_2) of the second image as follows:

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \quad (3.2)$$

If the geometric transformation can be expressed as a pair of separable functions, i.e., such as two consecutive 1D operations, then significant savings in time and memory will be obtained during the implementation [47]. **Our proposed registration method in this thesis is based on this concept to decrease the complexity of the algorithm and consequently increase the efficiency of our method, wherever it is possible.**

3.3 Mutual Information Based Registration

Image registration is particularly difficult when images are obtained through different sensor types (multi-modal image registration), since the relationship between the images is complex and unknown. In 1995 Viola [45] proposed employing information theory to solve this difficult problem. The advantage of using information theory is that it provides sufficient flexibility for capturing

³<http://en.wikipedia.org/wiki/Image-registration>

the underlying relationship between the images [40]. Many robust and accurate multi-modal registration algorithms have been developed based on their approach, and Maintz et al. have surveyed some of these algorithms [39]. Although, the information-theoretic approach has been very successful in multi-modal image registration, speed has been always a critical issue because of the computational complexity of their similarity measure and/or high-dimensional optimization problem. Moreover, most of the information-theoretic registration algorithms discard spatial information in the images which may result in low accuracy [40], [41]. In the following sections we will give some background on the basic concepts of information theory, motivate their employment for alignment, and explain our proposed solution to solve the drawbacks related to Maximization of Mutual Information (MMI) based methods.

3.3.1 Information entropy and uncertainty

Hartley [24] defined the first information measure as:

$$H = n \log(s) \quad (3.3)$$

where n is the length of the message and s is the number of possible values for each symbol in the message. Assumes all symbols equally likely to occur.

Shannon’s 1948 paper, entitled “A Mathematical Theory of Communication,” [42] is widely accepted as the birth of Information Theory. Shannon uses probability theory to model information sources. The data produced by a source is considered as a random variable. The information content (Shannon’s entropy) of a discrete random variable X that has a probability distribution $p_X = (p_1, \dots, p_n)$ is then defined as:

$$H(X) = H(p_X) = \sum_{i=1}^n p_i \log(1/p_i) \quad (3.4)$$

The base of the logarithm determines the unit, e.g. if base is 2 the measure is in bits. Thus, entropy is merely a statistical average of uncertainty or

information.

3.3.2 Entropy of an image

To calculate the entropy of an image, we can analyze the image as a high dimensional discrete random variable and use the same equation (3.4). Also, the entropy of the image can be calculated using its histogram as follows.

The normalized histogram is the probability of pixel intensities:

$$p(i) = h_I(i)/N \quad (3.5)$$

where $h_I(i)$ is the histogram value of intensity i in image I and N is the total number of pixels of image I . Using this model, the entropy of the image can be computed as:

$$H(I) = \sum_i h_I(i) \log N / h_I(i) \quad (3.6)$$

3.3.3 Joint Entropy

As it is mentioned before, entropy is the amount of information an event provides. Another interpretation of the entropy is the dispersion in the probability distribution. For example an image of a single amplitude has a less disperse histogram than an image of many greyscales. So, the lower dispersion implies lower entropy [3]. Based on this fact, entropy can be used for image registration as follows. First, a joint probability distribution (histogram) of two images should be generated. A joint histogram of two images is a two-dimensional density function containing the number of simultaneous occurrences of intensities between the two images. In other words, in a 2D joint histogram each axis is the number of possible greyscale values in each image. Each histogram cell is incremented each time a pair $(I_1(x, y), I_2(x, y))$ occurs in the pair of images [3]. Intuitively, we expect the dispersion to be minimal when the interdependence of the grey-values of both images to be registered is maximal. Thus, when the images are perfectly aligned, the projection of the density function will be on an identity-line from the minimum intensity to the maximum intensity. As

the images mis-align, the dispersion grows. Thus, two images are registered when the dispersion in their joint histogram is minimized. Mathematically, for registration, joint entropy (equation 3.7) must be minimized:

$$H(I_1, I_2) = - \sum_{i,j} p(i, j) \cdot \log[p(i, j)] \quad (3.7)$$

3.3.4 Mutual Information

One of the definition of mutual information is:

$$MI(I_1, I_2) = H(I_1) + H(I_2) - H(I_1, I_2) \quad (3.8)$$

Based on this definition, maximizing the mutual information is equivalent to minimizing the joint entropy (last term).

Advantage in using mutual information over joint entropy is that it includes the individual inputs entropy as well which makes it work better than simply joint entropy in regions of image background (low contrast) [3].

Maximization of MI is a very general and powerful criterion because no assumptions are made regarding the image content of the modalities involved which makes it suitable for multi-modal image registration.

3.3.5 Our Modifications to MMI

In spite of the fact that mutual information (MI), as a similarity measure, has been very successful in multi-modal image registration, the iterative nature of the method for finding optimal transformation makes it time consuming and mis-registration can happen due to local extrema. We propose a heuristic approach based on straight line detection and matching for rigid registration. Our proposed method improves efficiency and performance of the MI-based registration methods by converting the 3-D optimization problem into a 1-D search along the corresponding lines [50, 51]. The proposed method is suitable for images rich with line segments, such as those used in remote sensing applications. Rigid transformation is described as equation 3.9.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (3.9)$$

We determine the rotation parameter θ of the optimal rigid transformation in (3.9) by finding a pair of corresponding lines on both images with a high confidence similarity score. First, line segments are detected using Canny edge detector [14] and Hough transform [19]. Then, Mean Standard Deviation Line Descriptor (MSLD) [46] is employed to find the similar pair of lines (Look at appendix C for details about this descriptor). We then use this pair to achieve translational parameters t_x, t_y by maximization of MI.

Finding corresponding lines as the starting step of the algorithm has two advantages: (a) The method is more efficient as we only search along the similar lines detected on both images (1-D search instead of N-D optimization) (b) performing 1-D exhaustive search along the corresponding lines helps to skip local minima which can mislead the registration method. Flowchart in figure 3.1 shows the details about the proposed method.

3.4 Why not MMI?

Our proposed MMI-based registration method performs well when the images have identifiable corresponding features and high quality, which was the case for our initial small set of experimental images. However, when our set of images was extended, we encountered problems that MMI-based methods can not handle, due to the low quality of images. Various factors affect the quality of both satellite and street-level fisheye images and eventually cause the result of intensity-based methods, such as our proposed MMI-based method, to be unreliable. Some of these influential factors are as follows:

- Differences of view-points between satellite and fisheye images
- Distortion of fisheye images
- Presentation of different objects in the images that do not have correspondence in the other image, such as vehicles, pedestrians and lamp

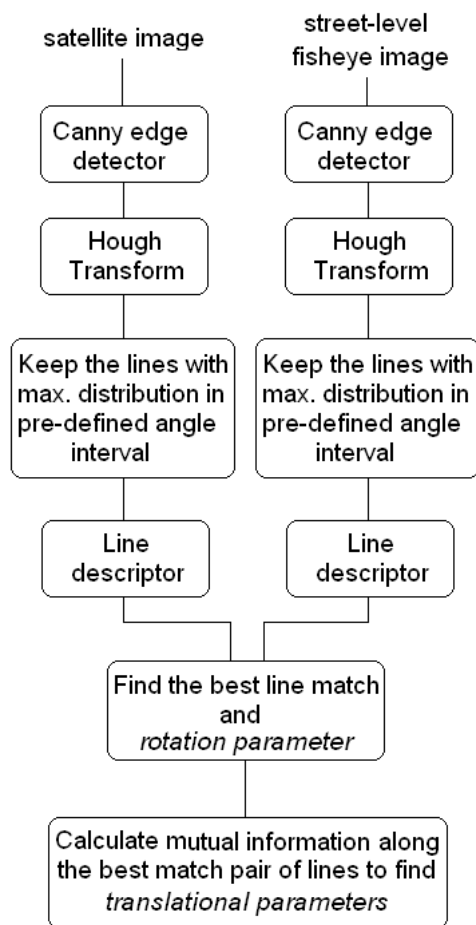


Figure 3.1: Main steps of the proposed MMI-based registration technique

posts.

- Occlusion of corresponding features in images. For example branches of trees can occlude the driveways in satellite images.
- Differences in lighting and atmospheric conditions between images. The main issue here is shadows that can cause the similar intensities look different in corresponding images.

For example, Figure 3.2 shows that MI score for non-corresponding images could be as high as corresponding images. In this figure, the image shown at the top is part of satellite image containing a road, and the three images shown

at the bottom are fisheye images. The first two fisheye images do not contain any roads. Only the third fisheye image contains a road. However, the MI score between the satellite image (top image) and each of the fisheye images at the bottom is 0.5. The reason is that shadows and pedestrian generate road-like features in first and second fisheye images.

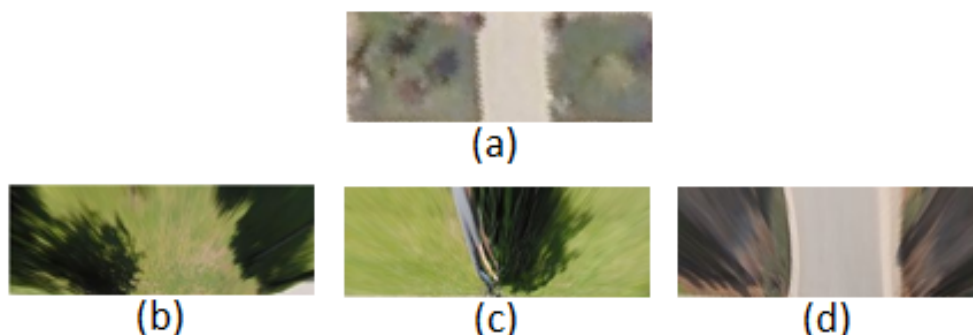


Figure 3.2: Example of failure of MMI-based registration methods. (a) Part of satellite image. (b),(c) Fisheye images with road-like features. (d) Fisheye image with real road. MI score between (a) and ((b) or (c) or (d)) is the same.

MMI-based methods are intensity-based registration methods that rely on finding a relationship between intensity values of corresponding pixels of images to be registered. However, due to the above issues, this relationship does not always exist. Therefore, this problem should be addressed so that a relationship can be found between the images regardless of the situation. We found that using Chamfer matching as the similarity measure is a good starting point for developing an appropriate registration method for our images. The most important advantage of the Chamfer matching method is its ability to handle imperfect (i.e. noisy, distorted, etc.) data, as it does not need an exact match between the features. Before explaining our final proposed method, in chapter 5 we explain the details about our modified version of the Chamfer matching algorithm which is basically the core of our algorithm.

Chapter 4

Data Preparation/ Description

In this chapter, the process of preparing satellite and street-level fisheye images will be explained. Fisheye images are collected from the streets and processed to extract the top-view of each fisheye image. In order to have more unique features, we work with stripes instead of individual fisheye images. In this chapter, we explain how the fisheye stripes can be made by seamlessly stitching individual fisheye images (section 4.1). Also, generating satellite stripes using Google map blocks will be explained in section 4.2.

4.1 Fisheye stripe

To stitch two images together in a seamless manner, a method proposed by V. Kwatra et al. [28] was employed. They used s/t graph-cuts (see Appendix B) to find the best cut in the overlapped area of the two images, as will be explained further below.

Suppose we want to stitch together overlapped images $I1$ and $I2$ (see Figure 4.1). The chosen path is through the pixels wherein the colors of $I1$ and $I2$ are exactly, or approximately, identical. This path determines which image contributes pixels at different locations in the overlapping region.

Let p and q be two adjacent pixel positions in the overlapping region. Also, let $A(p)$ and $B(p)$ be the pixel colors at position p in $I1$ and $I2$, respectively. The matching quality cost (i.e. capacity of the link in the graph) W between

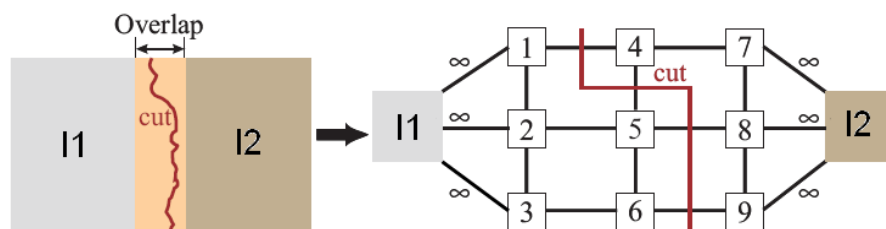


Figure 4.1: Stitching two images can be converted to an optimization problem to be solved by graph cuts. Image courtesy of V. Kwatra et al. [28]

the two adjacent pixels p and q is defined to be [28]:

$$W_{pq} = \|A(p) - B(p)\| + \|A(q) - B(q)\| \quad (4.1)$$

where $\|\cdot\|$ denotes an appropriate norm. Graph cuts can be applied to solve the path finding problem, if the capacity of all of the links in the graph are determined [28].

Figure 4.2 shows a fisheye stripe made by stitching seven fisheye images using the method explained above.

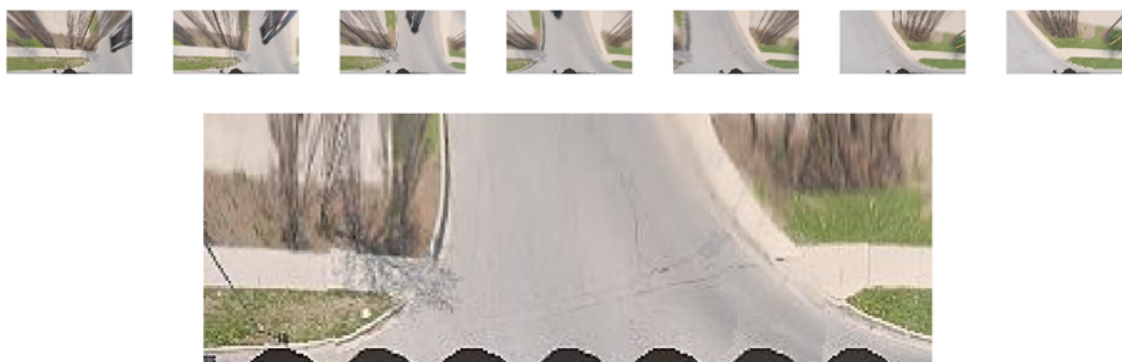


Figure 4.2: Seven fisheye images (shown in the top row) were stitched together seamlessly to make a fisheye stripe (shown in the bottom row).

4.2 Satellite stripe

To optimize the performance of map retrieval and display, the entire map of the world is cut into tiles of 256×256 pixels each. As the number of pixels differs at each level, so does the number of tiles: the number of tiles in the height and width of the map in each resolution is 2^{level} . Each tile is given a pair of coordinates ranging from $(0, 0)$ in the upper left to $(2^{level} - 1, 2^{level} - 1)$ in the lower right. The area that is covered by each tile, as well as the zoom levels (i.e. resolution in meters per pixel), are predefined for the whole Earth. There are three main systems of tile addressing¹: Google XYZ, Microsoft QuadTree, and TMS (Tile Map Service). The only difference between these systems is the way in which the equivalent tiles are indexed. Microsoft QuadTree is the system we have used in this thesis. In this system, the two-dimensional tile coordinates are converted into one-dimensional strings called “quadkeys”. Quadkeys provide a one-dimensional index key that usually preserves the proximity of tiles in XY space. In other words, two tiles that have nearby XY coordinates usually have quadkeys that are relatively close together. This is important for optimizing database performance, as neighboring tiles are usually requested in groups and it is desirable to keep tiles on the same disk blocks in order to minimize the number of disk reads.

Each quadkey identifies a single tile at a particular level. The length of a quadkey shows the zoom level of the corresponding tile. The quadkey of any tile starts with the quadkey of its parent tile (i.e. the containing tile at the previous level). As shown in Figure 4.3, tile 2 is the parent of tiles 20 through 23 and tile 13 is the parent of tiles 130 through 133.

The letters “q”, “r”, “s”, and “t” are also being used to code x and y quadrants in a pyramid of tiles, as shown in Figure 4.4. Substituting the letters with a binary scheme provides an approach for calculating neighboring tiles. Note that the Google image universe starts at quadrant t.

¹<http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/>

²<http://msdn.microsoft.com/en-us/library/bb259689.aspx>

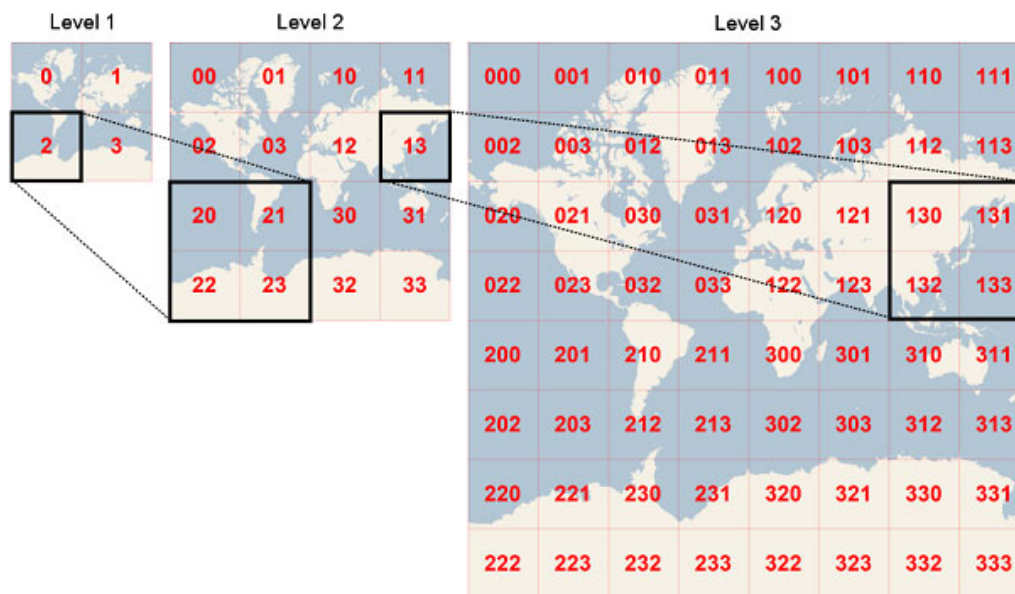


Figure 4.3: Quadkeys- binary representation: Tile system of the satellite image of the whole world in different resolutions. Image courtesy of Microsoft²

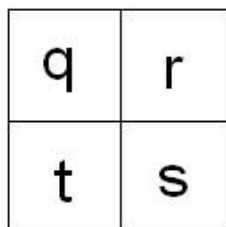


Figure 4.4: Quadkeys- letter representation.

Google maps API are available for downloading small tile map images from Google Maps, by inputting the longitudes and latitudes of the desired region. We used these APIs to develop a program that can download the tile images of a specific chain. The downloaded tile images can be combined, using their quadkeys, to form a large map. The original GPS data must also be converted into the new coordinates of this map. The geographic coordinates (latitude and longitude) can be converted into pixel coordinates as follows. Given latitude and longitude in degrees and the level of resolution, the pixel

XY coordinates can be calculated using equations 4.2 - 4.4:

$$L = \sin\left(\frac{\pi}{180} \text{latitude}\right) \quad (4.2)$$

$$X = 256 \left(\frac{\text{longitude} + 180}{360} \right) 2^{\text{level}} \quad (4.3)$$

$$Y = 256 \left(0.5 - \frac{\log\left(\frac{1+L}{1-L}\right)}{4\pi} \right) 2^{\text{level}} \quad (4.4)$$

The customized satellite map for each chain is made from the appropriate bitmap tiles of satellite images obtained from Google maps and are put together using their *quadkeys*.

After making the satellite map of a chain out of the Google map tiles, satellite stripe can be made by extracting a rectangular area around the location we want to fix. The rectangular area is determined using the original coordinates of the first and last fisheye images in the fisheye stripe. Also, using these coordinates the extracted rectangular area will be rotated to form the satellite stripe (look at Figure 4.5).



Figure 4.5: Part of a sample chain and the extracted satellite stripe

Chapter 5

Modified Chamfer Matching

Chamfer matching was first proposed in 1977 [7]. It is a high-level matching method that takes not only the features into account, but also the structural/spatial relations between them. In this method, the best fit of edge points from two different images can be found by minimizing a generalized distance between them. To calculate distance between two sets of edge points, a *distance transform* is used. A distance transform is the operation of converting a binary edge image to a distance map by giving the value of zero to the edge pixels and non-zero to non-edge pixels. Each non-edge pixel is given a value that is a measure of the distance to its nearest edge pixel. We have used Euclidean distance metric in our algorithm, as shown in Figure 5.1.

For the Chamfer matching method, the edge points of the second image (e.g. fisheye stripe) are superimposed on the distance map of the first image (e.g. satellite stripe). The measure of correspondence between the edges is called the *Chamfer matching score* and is given by:

$$CM = \frac{1}{n} \sum_{f_i \in F} \min_{s_i \in S} |f_i - s_i| \quad (5.1)$$

where $F = \{f_i\}$ and $S = \{s_i\}$ are the sets of edge points for fisheye and satellite stripes respectively. n is the total number of edge points in fisheye stripe. Basically, Chamfer matching score CM is an average of the values of the distance map of the satellite stripe along the edges of the fisheye stripe.

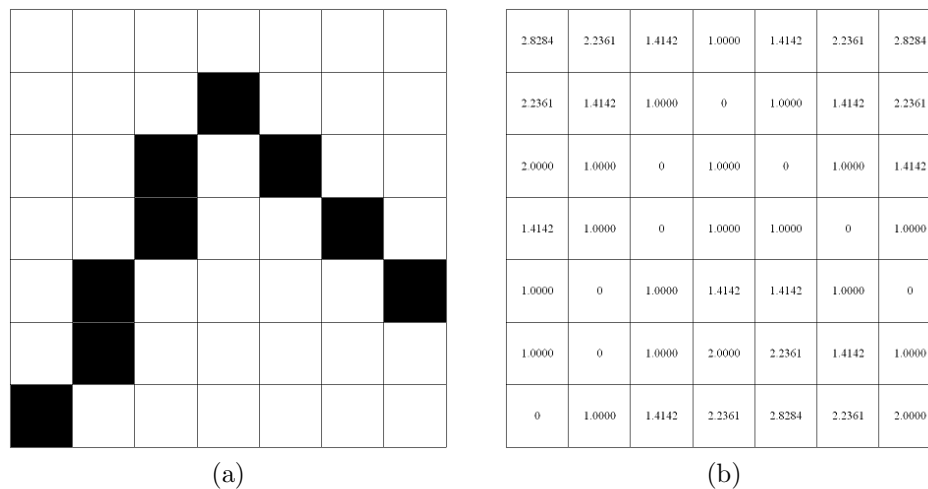


Figure 5.1: An example of (a) a binary image and (b) its distance map.

A perfect fit between the two edge maps will result in a Chamfer matching score zero. However, in reality the actual matching consists of minimizing this score by moving the fisheye stripe over the satellite stripe and finding location which has the minimum CM score [10]. The most important advantage of the Chamfer matching method is its ability to handle imperfect (i.e. noisy, distorted, etc.) data. However, it becomes less reliable in the presence of background clutter.

To improve reliability and accuracy of the standard Chamfer matching method, two modifications are made to equation 5.1 which will be explained below.

5.1 Integration of edge orientation information

Using only edge information is not always enough for matching. In the presence of background clutter, standard chamfer matching becomes less reliable [33], as it does not have any information about the image content. Integration of edge orientation proposed by M.Y. Liu et al. [33] can improve the robustness of

the standard Chamfer matching method by considering the image content as well. Therefore, the new score is given by adding another term to the standard Chamfer matching score in equation 5.1 as follows:

$$OCM = \frac{1}{n} \sum_{f_i \in F} \min_{s_i \in S} |f_i - s_i| + \lambda | \theta(f_i) - \theta(s_i) | \quad (5.2)$$

where the first term is the location term (same as equation 5.1), the second term ($| \theta(f_i) - \theta(s_i) |$) is the orientation term, and λ is a weighting factor between location and orientation terms. λ is experimentally selected to be equal to 5 for our experimental images. Orientation term is the circular difference between the gradient directions in both images at the edge points.

5.2 Integration of edge strength

Detecting correct edges directly affect the performance of the Chamfer matching algorithm. We identified the edges of a satellite stripe by segmenting it into two different classes of foreground (e.g. roads, driveways and sidewalks) and background. To find the best threshold for this segmentation, we used Otsu's method [38]. Otsu's method automatically finds the optimum threshold to convert a gray-level image to a binary image by maximizing the interclass variances and minimizing the intraclass variances. Otsu's method assumes that the image to be thresholded contains two different classes of pixels (e.g. foreground and background). Threshold that Otsu's method finds is small for some chain of our images. For example, as Figure 5.2 shows, in Otsu's method a tree or grass on the right hand side of the image are considered as part of the foreground. However, we are only interested in detecting the edges that are identifiable in both fisheye and satellite images, such as roads and driveways. Therefore, to eliminate objects other than roads and driveways, as much as possible, we added an offset to the Otsu's threshold for some of the chains. This offset was experimentally set for each chain. We also cleaned up the binary image by excluding any small objects, using morphological opening. The morphological open operation is an erosion followed by a dilation, using

a single structuring element [22]. We used a disk-shaped structuring element with radius 1 (pixel).

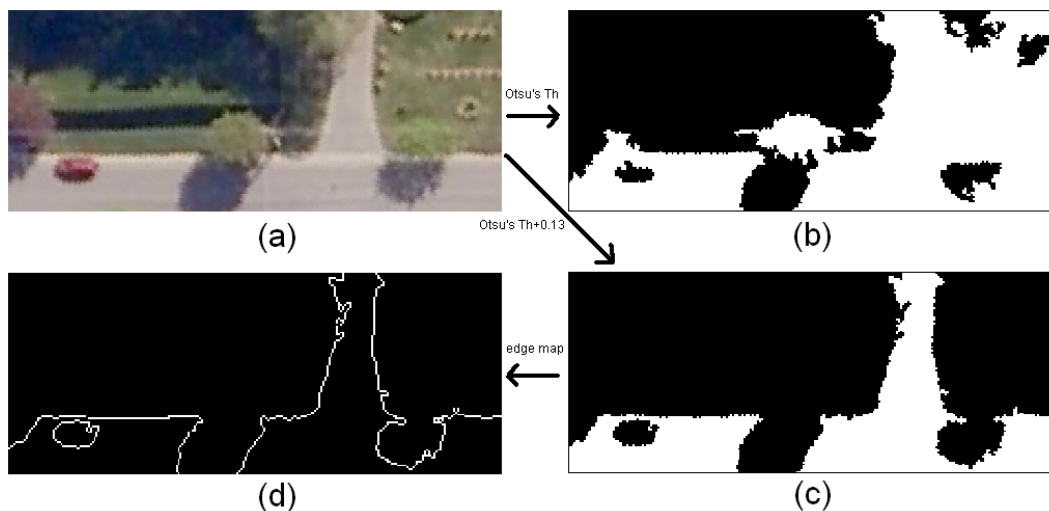


Figure 5.2: Construction of the edge map for the satellite stripe: (a) a sample satellite stripe (b) binary image obtained using Otsu's method (c) binary image obtained using threshold = Otsu's threshold + 0.1 (d) edge map

Finding the edges of street-level fisheye images is even harder than satellite images due to the high variations of intensities, objects and distortion. Figure 5.3 (first and second rows) shows some sample street-level images that Otsu's method did not work well for them. Also Figure 5.3 (third row) shows our effort to improve the results of Otsu's method by adding an offset to the Otsu's threshold (like what we did for the satellite images). However, this is not practical, because to get the best result we had to set the offset for each of the images of a chain individually. More importantly, in some of these images (like the 1st and 2nd column) it is impossible to separate the roads from the rest of the image even by carefully setting the offset. The reason is that the intensity histogram of these images is not separable at all into two modes (foreground and background).

In order to avoid missing any correct edges and also alleviate the effect of false edges on the final result, we augmented the strength of the edge points into the Chamfer matching score. Therefore, our new modified Chamfer

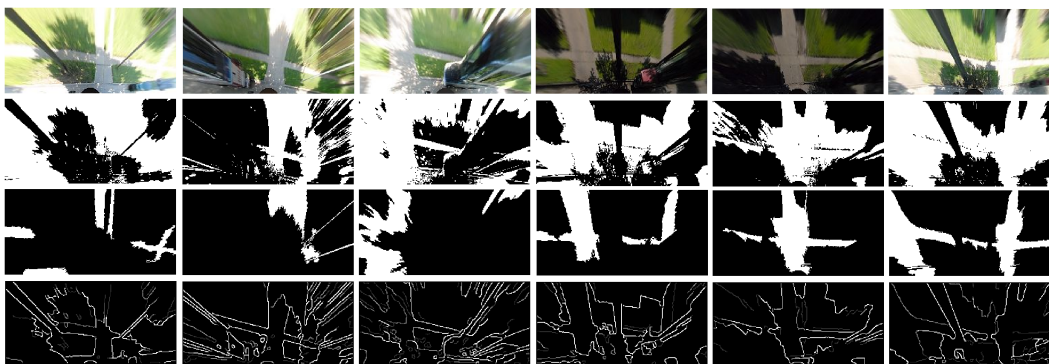


Figure 5.3: (1st row): sample fisheye images from one chain. (2nd row): binary conversion using Otsu’s method. (3rd row): binary conversion using Otsu’s threshold+offset(0.3, 0.3, 0.3, 0.15, 0.1, 0.25 respectively). (4th row): our proposed edge map based on multi-thresholding.

matching score is given by:

$$MCM = \frac{1}{n} \sum_{w_i, f_i \in F} \min_{s_i \in S} (w_i |f_i - s_i| + \lambda |\theta(f_i) - \theta(s_i)|) \quad (5.3)$$

where the weight w_i is high for the strong edge points and low for the weak edge points. This added weight encourages the minimization algorithm to locate the strong edges on the low distances on the distance map of the first image (e.g. satellite image).

To calculate these weights, we used three different thresholds for the Canny edge detector [14] to make three different edge images. The final edge map is the sum of these three edge images. As the strong edges appeared in all the edge images, they will have higher value in the final edge map. Figure 5.4 better clarifies the method on a sample image. Also, Figure 5.3 (4th row) shows the edge maps of the sample images obtained using the explained method.

As we will see in chapter 7, based on the limited set of experiments we found that this method leads to more accurate results compared to the standard chamfer matching, because regardless of existence of strong “false” edges, the “correct” edges still have a chance to appear in the edge map. However, we did not perform any formal evaluation.

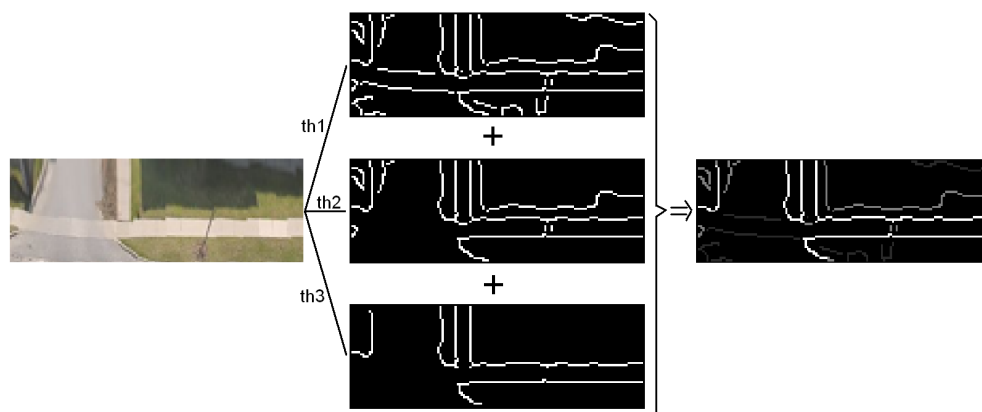


Figure 5.4: Construction of the multi-level edge map for a sample fisheye stripe.

Chapter 6

Methodology

In this chapter, we will explain our proposed algorithm for automatically and accurately geotagging street-level fisheye images. In our algorithm the geotagging task is performed by detecting and aligning similar features between the satellite and fisheye images. These features must be something that are identifiable on both satellite and fisheye images, such as roads, driveways and sidewalks. As Figure 6.1 shows, the algorithm composed of three modules: image selection, registration and evaluation. The algorithm starts by searching through the street-level fisheye images of a chain and finding those images containing features that can be used for registration. The “image selection” method that we have developed will handle this important task. Section 6.1, contains the explanation of this method in detail. After selecting the appropriate images, we extend them into stripes and registration process is performed on the stripes. Making the stripes from individual images was explained in chapter 4. The registration method is explained in detail in section 6.2. We have also designed an evaluation method to quantify the quality of the registration results and reject the weak results. We do that by measuring the similarity between the fisheye stripe and the final satellite image cropped from the satellite stripe (search area). This evaluation method and our similarity measure will be explained in section 6.3.



Figure 6.1: Overall view of the proposed method

6.1 Image Selection

In our algorithm, we used small roads and driveways that were perpendicular to the direction of the chain (i.e. appear vertical in the extracted stripe) as features. These features help us locate the shift along the chain. Because of this requirement, we searched within the fisheye images of the chain and selected the images that had such roads or driveways. To detect vertical roads/driveways, we matched the fisheye images against a model image shown in Figure 6.2. Matching against models is one of the well-known registration techniques [13]. We calculated the mutual information score for each fisheye image with this model image. If the score was high enough, there was a good chance of having a road or a driveway in the fisheye image. The threshold was selected to be low in order to catch most of the fisheye images that contained vertical features. We observed that a value of 0.3 performed reasonably well over a wide range of images.

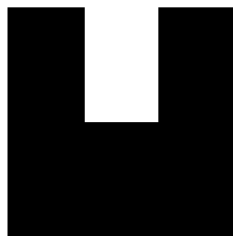


Figure 6.2: Model image to detect the vertical features on fisheye images

6.2 Registration method

In this section, we explain our registration method. The block diagram in Figure 6.3 shows the main steps of the proposed registration algorithm. The algorithm starts by breaking the chain of images into a number of stripes, based on the images selected in previous step (section 6.1). The registration process is repeated separately for each stripe. After the preparation of stripes, a search is performed to find horizontal lines that are similar between the fisheye and the satellite stripes. Detection of such horizontal lines allows us to align these stripes in a direction perpendicular to the chain. If similar lines are detected, then a search needs to be done in a 2-D space of (a) shifts along the chain and (b) scales. However, if no similar lines are detected, then a 3-D space of (a) shifts that are perpendicular to the direction of the chain, (b) shifts along the chain, and (c) scales should be searched to find a solution.

The third step is our modified Chamfer matching algorithm, which locates the best matches between the satellite and fisheye stripes in the search area. For each pair of stripes, Chamfer matching score versus the 2-D or 3-D search space has multiple local minima. The purpose of the last step is to perform an optimization over all of these local minima (of all the stripes) to pick the correct one. Dijkstra's algorithm is deployed to search for the optimal path among the set of local minima of all the stripes by applying a geometrical constraint. This constraint comes from the fact that the shift along the chain for different stripes should be as similar as possible. Finally, by having the shifts along and perpendicular to the chain, the GPS data for individual fisheye images in the fisheye stripe can be corrected. The following are the explanation of the above steps in more detail.

6.2.1 Detection of Line Correspondences

As demonstrated in Figure 6.3, after the preparation of the satellite and fisheye stripes the horizontal lines that are similar are detected on each pair of stripes. The details of this step are shown in Figure 6.4. As outlined in this flowchart, the Sobel Horizontal Edge Detector [22] and Hough Transform [19] are em-

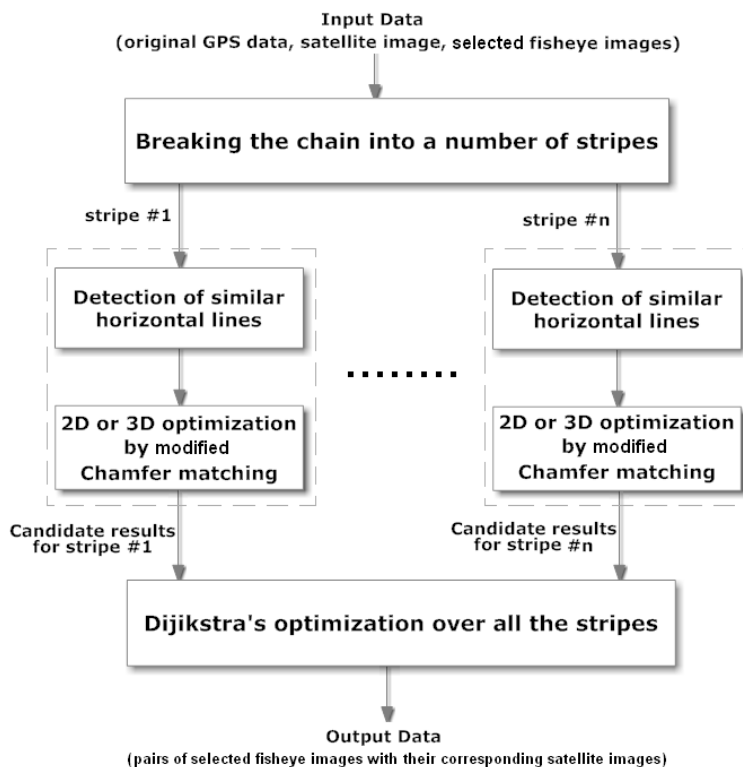


Figure 6.3: Overall view of the proposed registration method

ployed to detect horizontal lines on both fisheye and satellite stripes. Mean standard deviation line descriptor (MSLD) [46] is used to find the lines that are similar on both stripes. MSLD is a sift-like [34] descriptor for lines, which uses the gradients along different directions around the lines. We considered nine 5×5 sub-regions for each point on the line at the direction perpendicular to the line. In each sub-region, the gradients are accumulated in four different directions. By calculating the mean and standard deviation of the gradients for each row of sub-regions along the line, the line descriptor is obtained as a $72 - element$ vector (9 sub-regions $\times 4$ directions $\times 2$ statistical measures: mean, std). Further details about this descriptor can be found in Appendix C. By comparing the Euclidean distance between the MSLD vectors of the lines, we were able to find the most similar pair of lines on the satellite and fisheye stripes. When the angle difference between the most similar lines was

less than 20 degrees, the lines were considered correspondences.

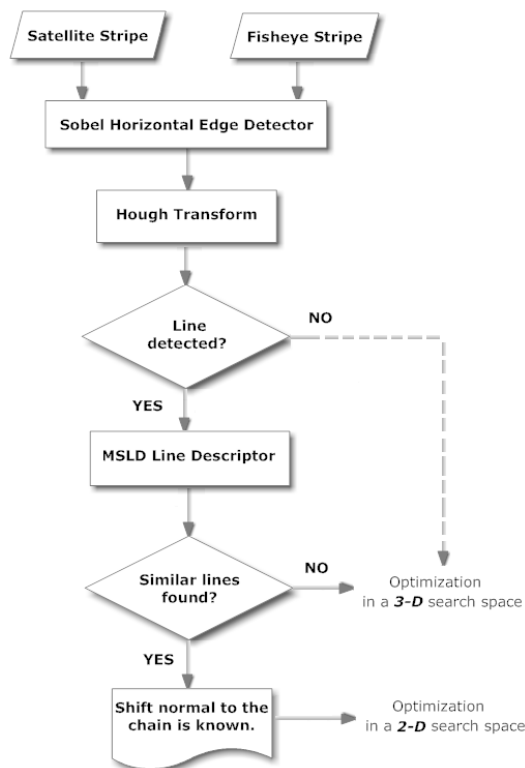


Figure 6.4: The process of detecting similar horizontal lines

It has to be mentioned that the purpose of this step is to reduce the size of the search space for Chamfer matching; this eventually results in speeding up the geotagging process. As demonstrated in Figure 6.4, if similar lines are detected, the search space for the Chamfer matching would have only two degrees of freedom instead of three.

6.2.2 Modified Chamfer Matching

The third step in Figure 6.3 is the optimization step using our modified Chamfer matching method, which was elaborated in chapter 5. The $2D$ (or $3D$) surface of the modified Chamfer matching score versus x , (y) , and $scale$ is not a smooth, convex surface. In fact it has a lot of minima, due to “false” edges and noise. For example, images may contain objects other than what we are

interested in, such as shadows of trees, lamp posts, vehicles, pedestrians, etc. These objects generate false edges in the edge map of the satellite and fisheye images. We call them false edges, because they are not identifiable on both satellite and fisheye images, and therefore can not be used for the purpose of matching. Later, these false edges generate local/global minima on the Chamfer matching score surface. The false edges can even be stronger than the edges we are interested in. So, considering the global minimum as the final result is not accurate, as any of the local minima could be the correct solution. Therefore, to prevent the removal of the correct answer, all the local minima must be kept for the next stage. To find these local minima, an unconstrained nonlinear optimization based on simplex search method [29] was employed.

6.2.3 Optimizing Stripe Alignment using Dijkstra's Algorithm

In this section, we explain the last step in Figure 6.3, which is the optimization among all of the stripes of a chain. In the previous step (the Chamfer matching), we found a number of local minima that were the candidate results for each stripe. In this step, we pick the correct local minimum by formulating another optimization problem. We used the criteria that all of the stripes in a single chain should have (almost) the same shift along the chain. This is guaranteed by the fact that fisheye images on a single chain were taken at equidistant intervals ($\simeq 15\text{feet}$). Therefore, for each stripe we have to find a local minimum that is compatible with the local minima associated with other stripes in the chain. We formulate this by constructing a graph using the local minima of the stripes as *nodes* and the difference between the shifts along the chain as *weights* as shown in Figure 6.5. In this graph, A and B are origin and destination nodes respectively and the S nodes represent local minima. The first and second indices of each node S are the stripe number and local minimum number in that stripe, respectively. For example, nodes S_{11} , S_{12} , and S_{13} are the three local minima of the first stripe. The weights of the links between different nodes set as follows:

$$\forall i, j : W_{ij}^A = 0, W_B^{ij} = 0 \quad (6.1)$$

$$W_{(i+1)k}^{ij} = |S_{ij}(\text{along_shift}) - S_{(i+1)k}(\text{along_shift})| \quad (6.2)$$

where $W_{(i+1)k}^{ij}$ is the weight between j -th local minimum of the i -th stripe and k -th local minimum of the $(i+1)$ th stripe. $S_{ij}(\text{along_shift})$ is the shift along the chain related to the j -th local minimum of the i -th stripe.

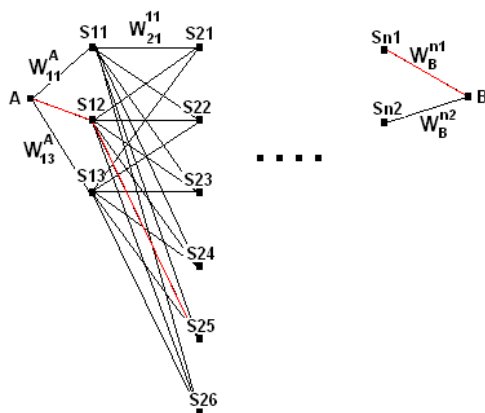


Figure 6.5: Finding the optimal path between the local minima using Dijkstra's algorithm.

After setting all of the weights, a dynamic programming method, such as Dijkstra's algorithm [18], can be used to find the optimal path between A and B . The red links in Figure 6.5 is an example of such a path. The optimal path has the minimum total weight along all of the other paths. Each stripe has one local minimum on the optimal path, which is the final result for that stripe.

6.3 Rejection of Weak Registration Results

The final results of the registration algorithm may contain error for the following reasons:

- In image selection step (section 6.1), fisheye images with no real features maybe selected as the candidates by mistake. For example Figure 6.6 shows that shadow could generate road-like features that lead to selection of those fisheye images by mistake.
- The corresponding features (e.g. roads, driveways) on the search area of the satellite image may be occluded by trees, shadows, etc. For example Figure 6.7 shows two satellite stripes and their corresponding fisheye stripes. In the top row, the driveways are occluded by trees, shadows and vehicles which makes it hard for the algorithm to detect them. In the bottom row, as the satellite image shows, the area is under construction and the edges of the driveways are not identifiable.



Figure 6.6: Image selection algorithm may fail, due to shadows that generate road-like features.

To reject the wrong results, the similarity between the fisheye and final satellite stripes is calculated, using the normalized cross correlation (NCC) similarity measure. As we are interested in comparing the fisheye and final satellite stripes regarding the road patterns, we calculate NCC on the thresholded binary images instead of intensity images. Also, to decrease the sensitivity of the method to the threshold, we combine the binary images with different values of threshold. All the thresholds are set experimentally. There is no need to carefully tune the values for these thresholds, as we only want to catch the road pattern if exists (besides other false objects). However, we have not done any formal evaluation for parameter sensitivity. Using trial and



Figure 6.7: Satellite stripes and their corresponding fisheye stripes. top row: driveways are occluded. bottom row: driveways are not identifiable due to construction.

error, we found out that in order to remove those images containing error more than 2.5 meters, any result with $NCC \leq 0.2$ has to be ignored. Figure 6.8 shows the details of this evaluation method for a sample pair of images.

Applying NCC on original images does not work well, due to occlusion, shadows, non-flat objects such as trees, lamp posts, vehicles, etc. For example, for images in Figure 6.8 we should get high score for evaluation; NCC score for original images in this figure is 0.16, and for multi-thresholded images is 0.64. Therefore, NCC on multi-level thresholded images is a better choice, because it better reflects the quality of registration.

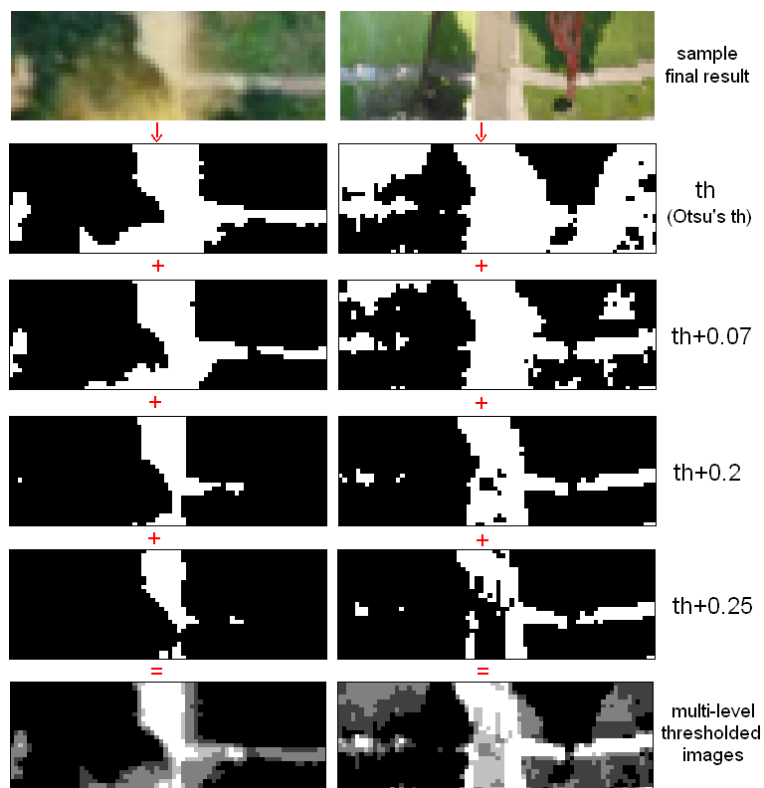


Figure 6.8: Evaluation of final result for a sample pair of images.

Chapter 7

Experiments and Results

In this section, we explain the experiments that we designed to evaluate the performance of our proposed geotagging method. Our test case is the top-view of street-level fisheye images of 20 chains in residential areas of north American cities (Appendix D includes the satellite images of these chains). The geotagging is done by finding the location of these images on satellite image (Google map) via our new image registration algorithm.

The preparation of data such as fisheye image stitching and tile extraction from Google map was done in *VC++* and *C#*. However, the registration algorithm itself is developed in MATLAB. Registering two stripes of images in MATLAB takes about 10 seconds on an Intel(R) Core(TM)2 Quad CPU computer @ 2.40GHz, 2.0GB RAM under Windows XP OS.

As explained in section 4, currently, the geo-spatial companies (such as iLook-about Inc.) fix the error of GPS longitude and latitude using odometry, and then human operators manually fix any remaining error that could be up to 6 meters. We use this manually-corrected locations as the ground-truth/gold-standard. To measure the error, we indicate the Euclidean distance between the manually corrected coordinates and coordinates determined by the proposed algorithm.

7.1 Experiment 1

In this experiment we evaluate the performance of the proposed algorithm. Also, we compare the results with the standard Chamfer matching algorithm. The algorithm is tested on 20 chains composed of more than 7000 fisheye images. From these fisheye images, 502 images are selected by our image selection algorithm (section 6.1) to be registered.

In table 7.1, the second column shows the number of images registered for each chain. The third column (E_{CM}) shows the average error for the standard Chamfer matching method, and the fourth column (E_{MCM}) shows the error for our modified Chamfer matching method. As it can be seen from the values of E_{CM} and E_{MCM} , when the quality of images are good in a chain, the errors are almost the same. However, for the chains with various issues (e.g. occlusion, background clutter), our modified Chamfer matching method shows 1 – 2 meters improvement over the standard Chamfer matching. This table also shows the average error after the evaluation (explained in section 6.3): The last two columns are the number of reliable results ($NCC > 0.2$) and the average error for them. As this table shows, after the evaluation, the average error for each chain is less than 2.5 meters. Also, the average error for all of the chains is 1.93 meters.

Note: Appendix E contains tables that show different results (including the error) for each stripe in each chain.

In Figure 7.1 some final results are shown visually. In each set the fisheye stripe and the final satellite image cropped from the satellite stripe are shown. Also, for comparison the original coordinates, manually corrected coordinates and the coordinates corrected by our proposed algorithm are shown in blue, red and green, respectively. As it can be seen from the results, the coordinates that our algorithm has found are close to the manually-corrected coordinates.

Chain Id	no. images	E_{CM} (m)	E_{MCM} (m)	no. reliables	E_{final} (m)
45614	11	2.09	2.01	9	1.49
45616	11	4.52	3.64	6	2.12
55764	8	1.94	1.84	8	1.84
55938	65	4.71	2.79	48	2.13
55964	28	4.64	2.59	24	2.22
93849	11	4.63	1.92	6	1.82
151323	20	3.13	2.65	17	2.39
151325	36	3.32	3.17	16	1.99
284553	39	4.18	2.76	22	1.96
286309	38	4.06	3.31	22	2.26
286313	11	3.64	1.95	8	1.94
287607	13	4.81	3.43	3	2.27
112357	62	3.27	1.59	51	1.38
114759	64	2.23	1.88	53	1.58
114761	20	1.87	1.13	12	1.06
230253	6	2.86	2.68	4	2.19
290663	12	4.94	3.26	4	1.08
53190	23	4.14	3.24	13	2.16
53456	17	3.05	2.96	9	2.19
54285	7	5.82	4.77	3	2.49

Table 7.1: Results: E_{CM} , E_{MCM} are the average errors from standard Chamfer matching method and our modified Chamfer matching method. E_{final} shows the final error of the algorithm after rejection of weak registration results using NCC .

7.2 Experiment 2

As mentioned before (chapter 6), the quality of images affects the results of registration algorithm (section 6.2). That was the reason that in the image selection step (section 6.1) we tried to select those fisheye images containing identifiable features. However, the question is how much effect the quality of images has on the final results? This experiment is designed to answer this question: it quantifies the overall impact of the quality of the input images on the performance of our proposed registration algorithm.

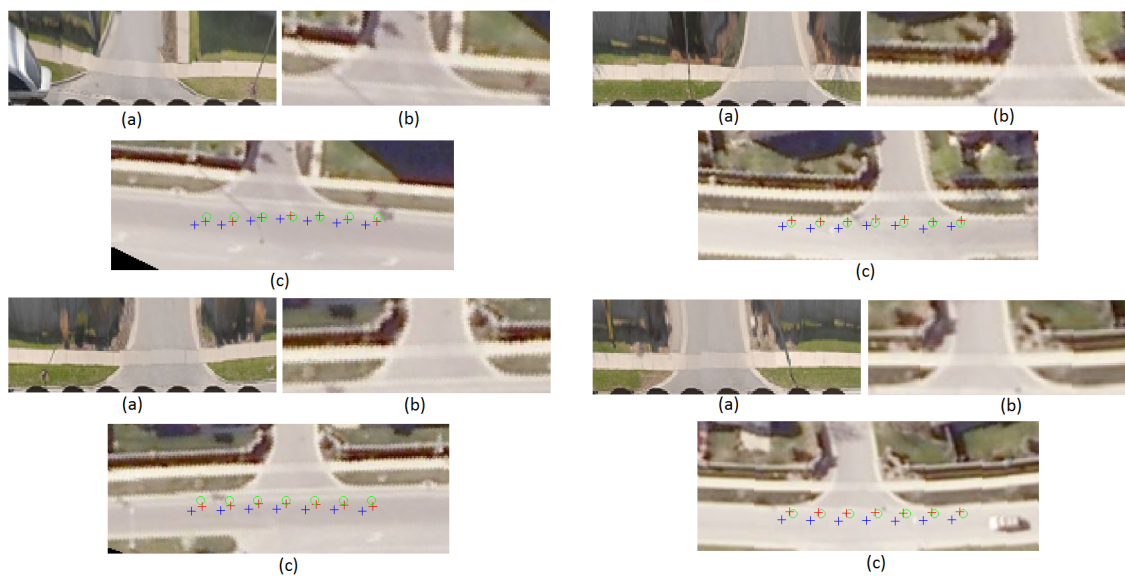


Figure 7.1: Some of the final results for $chainId = 45614$ (a) Fisheye stripe (b) Final satellite stripe (c) original, manually-corrected and algorithm-corrected coordinates shown in blue, red and green respectively.

Designing of this experiment is inspired by Monte-Carlo methods¹ which are a class of computational algorithms that rely on repeated random sampling to determine the distribution of the output as a function of inputs' distributions. In our case, the images can be considered as random variables and we want to determine the distribution of the registration error as a function of input images. Below we explain the details of this experiment.

Step 1: Contrary to experiment 1 (section 7.1) where the “image selection” algorithm (section 6.1) selected a number of images to be registered, in this experiment we run the proposed registration algorithm for **all** the images in each chain, and we record the error. The total number of images in all the 20 chains is 7801, and the algorithm could register 2322 of them. The rest of them were excluded by different constraints in the algorithm.

Note: the average error for the 2322 registered images was 2.35 meters. However, the average error for 1796 of them with $evaluation\ score \geq 0.2$ (section

¹http://en.wikipedia.org/wiki/Monte_Carlo_method

6.3) was 2.08 meters.

Step 2: As mentioned before, in the existing manual correction of GPS data, human operator only fixes the location of a few street-level images on satellite image, and the rest of the images will be located in between of these fixed points equidistantly. In this step of the experiment, we want to model the action of the human operator: out of all the results recorded in step 1, we randomly select 20 of them, and calculate the average error of them. We repeat this step several times (e.g. 100000).

Note: neither the number of selected images (20) nor the number of samples (100000) change the overall shape of the error distribution.

Step 3: By having the average error of 100000 samples, we can obtain useful statistical information. The best way to visualize this information is through the histogram. Figure 7.2 shows the distribution of error for those 100000 samples collected in step 2. As this distribution shows, the maximum probability belongs to $error = 2.33m$. Also, the probability of error being less than 2.5 meters is about 70%.

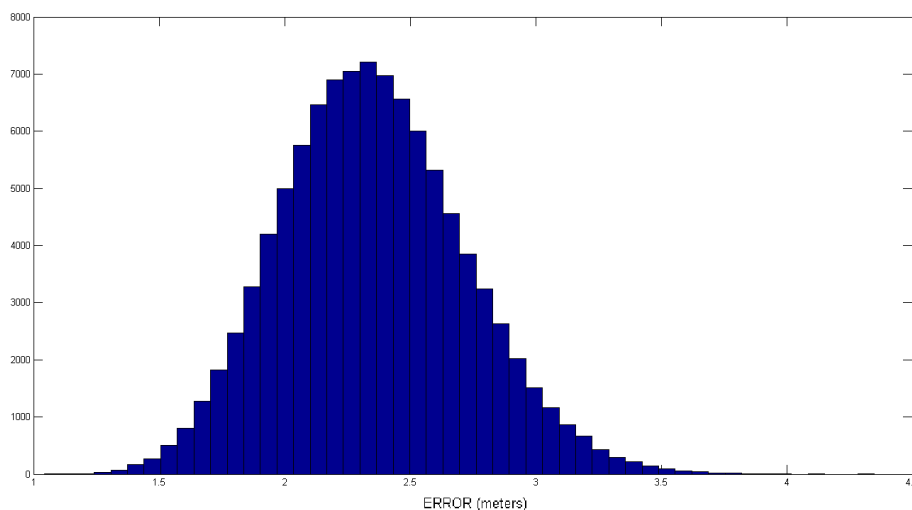


Figure 7.2: Error distribution of the proposed registration algorithm for all the images in 20 chains

Chapter 8

Software Architecture

The objective of this chapter is to explain the implementation of the proposed algorithm for automatic correction of geotag data of the street-level fisheye images, obtained from GPS. As Figure 8.1 shows, the inputs of the algorithm are original coordinates of the street-level fisheye images, satellite image of a chain and fisheye images of the chain. And, the output of the algorithm is corrected coordinates for each fisheye image.

Note: In Figure 8.1, the “original XY” is the coordinates of the street-level fisheye images calculated by converting the longitude and latitude obtained from GPS and odometry data fusion. As explained in the thesis these coordinates contain error and our algorithm which is based on image registration decreases this error. Therefore, the “new XY” are the coordinates of the street-level fisheye images of the chain with less error than the original ones.

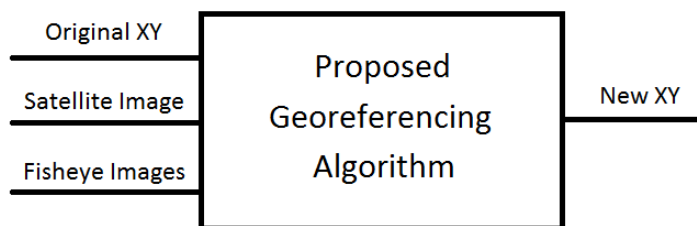


Figure 8.1: Functionality of the proposed algorithm

Figure 8.2 shows the overall architecture of the software and the flow of

data among different modules. As this figure shows, there are eight main modules that interact with each other and we explain them below:

- **Fisheye Stitcher:** This module stitches all the street-level fisheye images in a chain. Later, fisheye stripes will be extracted from this long stripe based on the output of the “Image Selector” module (look at “fish-eye stripe extractor” and its relationship with other modules in Figure 8.2).
- **Chain Satellite Image Maker:** This module is responsible for putting the Google map tiles together in order to make the satellite image of a specific chain.
- **Image Selector:** This module selects the street-level images which are appropriate for registration out of all street-level images of a specific chain. Appropriate images are those which contain features that can be identified on satellite images as well, such as roads and driveways.
- **Satellite Stripe Maker:** This module extracts a rectangular search area from the satellite image of a chain, based on the approximate location of street-level fisheye images obtained from GPS. Also, the extracted rectangular area is rotated to make a horizontal satellite stripe. The angle of rotation is obtained approximately using the GPS coordinates of the first and last fisheye images in a fisheye stripe.
- **Similar Horizontal Line Detector:** This module detects the straight horizontal lines on each pair of satellite and fisheye stripes, using Hough Transform. Then, it determines the most similar pair of lines based on the Euclidean distance of their MSLD vectors. The similar horizontal lines, if they are detected, will make the search area smaller as there is no need to search vertically anymore.
- **Modified Chamfer Matching:** This module forms the surface of the Modified Chamfer Matching score on a 2D or 3D search region and determines the local minima of this surface.

- **Dijkstra’s Algorithm:** This module finds the best local minimum out of all local minima found by MCM module, for each stripe. The module is employed Dijkstra’s algorithm to find an optimal path through different columns of local minima for different stripes in a specific chain. By determining the best local minimum, the location of each fisheye stripe on satellite stripe is determined, which results in declaring the more accurate geotag data for the fisheye stripes.
- **Evaluator:** This module is responsible to compare the fisheye stripe with the final satellite image cropped from the satellite stripe to determine their similarity. If the defined similarity score is high, those stripes are accepted as correspondences and the geotag data of the fisheye stripe is corrected accordingly.

Note: In Figure 8.2, (X_i, Y_i) represents the original coordinates of the location of street-level fisheye images in a specific chain that *Chain_Id* declares. These coordinates are obtained by converting the GPS longitude and latitude into the coordinates of the custom satellite image of that specific chain. Also, in Figure 8.2, I_s and I_f represent satellite and fisheye stripes respectively.

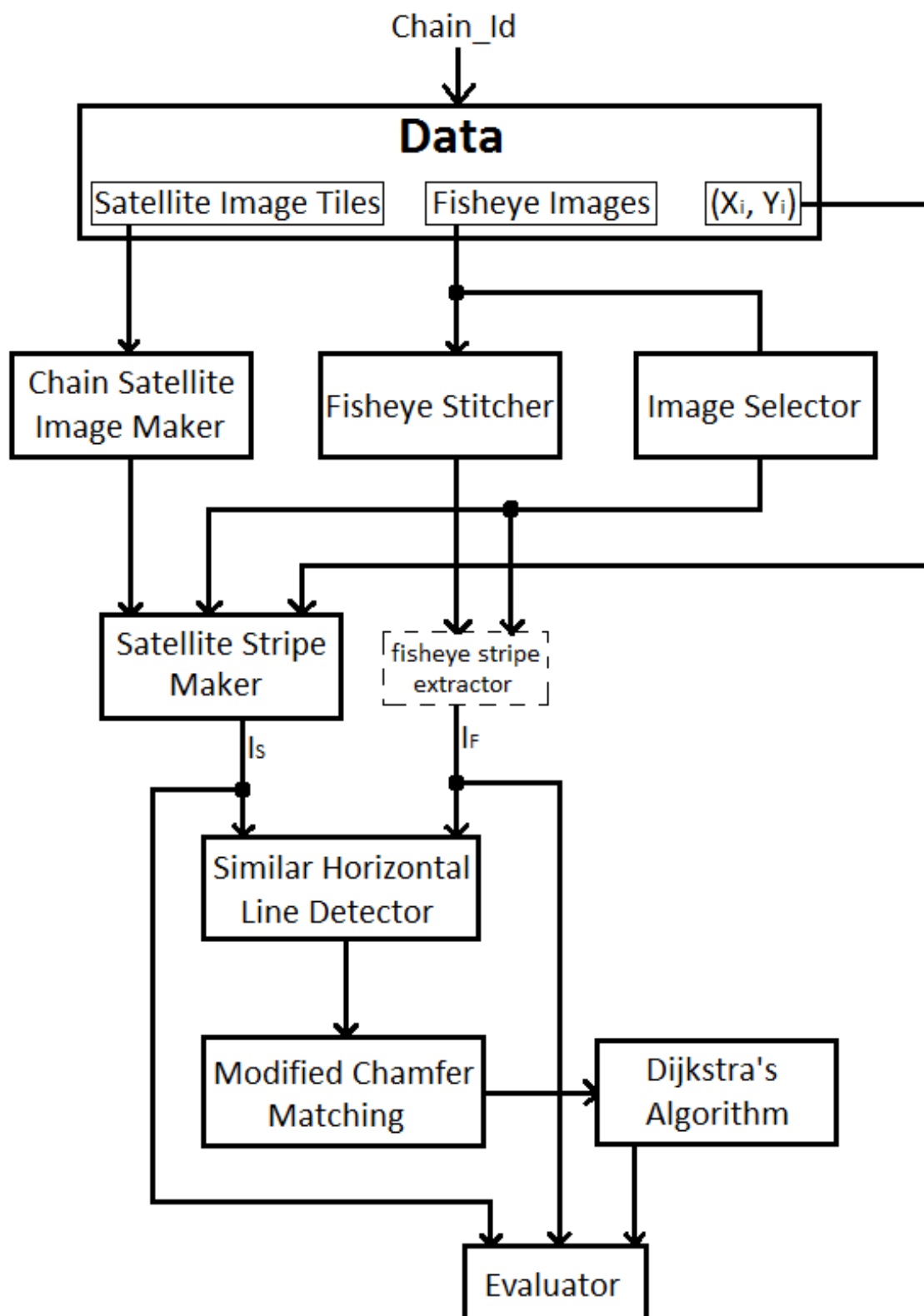


Figure 8.2: Overall architecture of the software for the proposed geotagging algorithm based on image registration

Chapter 9

Concluding Remarks and Future Work

Having accurate geotagged street-level images is essential for various businesses. We proposed a software-based method to correct the inherent error of GPS data for street-level images, which could be up to 15 meters. The proposed method was applied to correct the GPS data of images from 20 chains, and our experimental results show that the average error of 1.93 meters is achievable, if we select the appropriate images using the proposed image selection algorithm explained in section 6.1. We also statistically showed that if the images are selected randomly, the distribution of error has mean and standard deviation of 2.08, 0.32 meters, respectively. In this chapter, overall concluding remarks including discussion, contribution and suggestions for future work are given.

9.1 Discussion

Below we have a number of clarification about the algorithm:

- iLookabout Inc. fuses the original GPS data with the odometry data while collecting the images, and they only record that new data. Therefore, in our experiments, instead of original GPS data we had to use the

GPS data fused by odometry data which has less error (6 meters). However, our algorithm is not dependant on odometry data. We only need to make the search area big enough to cover the error of original GPS data. Considering smaller search area definitely increases the speed of the algorithm. It also makes the algorithm more reliable, because when the search area is smaller, less number of local minima is generated on the surface of modified Chamfer matching score.

- The proposed geotagging algorithm only uses the original GPS data to determine the search area, so the algorithm is applicable if the search area can be determined by other means. For example, by using the trajectory of road networks (or available road map vector) of a chain and the approximate location of the first image, the approximate location of the consecutive images can be obtained using their indices, that shows their order relative to the first image.
- In spite of the fact that we used manually corrected data as the gold standard, this data itself is not 100% accurate. As explained before, human operators fix the location of a few images based on their judgement. Correction along the chain is easier for the operators, as they will match features such as corners of grass or driveways, etc. However, performing correction perpendicular to the chain is not as easy as correction along the chain. Contrary to our proposed algorithm, operators do not really match the features pixel-based. Instead they only visually match horizontal features such as edges of the roads or sidewalks. Besides, the images that operators select to fix are not necessarily the same as the ones selected by the algorithm. Despite these facts, the only available gold standard, is the manually registered images.
- Our proposed algorithm is based on an optimization method, and it finds the location on the satellite image which has the maximum similarity with the fisheye image. Obtaining accuracy more than what was reported in chapter 7 may not even be possible due to inherent distortion of top-view of fisheye images. For a lot of businesses, such as the

clients of iLookabout Inc., accuracy of less than 2.5 meters is acceptable. Obviously, if more accuracy is needed, other solutions must be considered.

9.2 Thesis Contribution

The main contributions of this thesis are as follows:

- Our proposed geotagging method for chain of street-level images is a software-based method. Therefore, compared to its alternative existing sensor-based methods, our method does not need any navigation system be installed on the vehicle collecting the images. As a result, our method is a more economical way of geotagging and it can be applied on existing datasets because it only requires post-processing of collected data.
- Our proposed modified Chamfer matching method introduces a new set of weights to the objective function of the standard Chamfer matching score. This proposed set of weights increases the robustness of algorithm by undermining the weak false features. Moreover, it preserves the solution when the false features are stronger than the correct features.
- Our proposed Monte-carlo-inspired statistical analysis for quantifying the performance of the algorithm is a novel way to evaluate the overall performance of an image registration algorithm regardless of image quality.
- Our proposed evaluation method which applies NCC on multi-level thresholded images is novel and shows better performance than standard application of NCC on raw intensity images.
- Our proposed modified MMI-based registration method is a new way of combining feature-based and intensity-based registration methods. The proposed method improves the efficiency and reliability of the standard MMI-based registration method, by decreasing the dimension of the search space.

9.3 Future Research

Possible directions for future research in the topics of the thesis are discussed below.

- Any projection of a 3D scene into a wide-angle image results in distortion. For example, perspective projection distorts shapes of objects, or stereographic projections bend linear structures. As our proposed registration method is based on straight line features of the edges of the roads, driveways and pavements, perspective projection was a good choice for us. However, research can be done to evaluate the performance of the algorithm if other projections, that preserve the straightness of the lines in the scene, are employed. For instance, the projection presented by Carroll et al. [15] that can preserve both local shapes and straightness of lines.
- In the proposed algorithm, we tried to decrease the dimensionality of the search area by initially detecting a pair of line segments which are similar using standard Hough transform and MSLD. Research can be done about possibility of using features other than straight line segments, such as similar curves or regions.
- Non-flat objects such as lamp posts, vehicles, etc. are one of the reasons that generate false edges in the edge map of the street-level images. Research can be done about the possibility of detecting and removing these objects beforehand. For instance, using the stereo vision can be considered for this purpose. Because, in two consecutive images with known distance, non-flat features change when the shooting angle changes.
- One of the research possibilities is finding minimal number of images that need to be corrected in each chain in order to achieve a specific error level. One of the factors is (obviously) the length of the chain. The result of this research can lead to generating a “reliability measure” for each chain at the end of registration. Reporting this measure is useful, as

the company can repeat the correction process manually for the chains with low reliability measure.

- Exploring effect of different factors (such as quality of images) on reliability measure for each chain is another interesting research direction.
- Developing methods to quantify the quality of images is another important topic that can affect the performance of the proposed registration algorithm, because low quality images can be removed from the pipeline of registration.

Appendix A

GPS and Its Sources of Error

The GPS system can be divided into three basic segments [2]:

- Space segment (satellites)
- Control segment (control stations)
- User segment (GPS receiver)

The space segment consists of at least 24 satellites. The first satellite was brought to its orbit in 1978.

The control segment is composed of a “master control station”, an alternate master control station, and a host of dedicated and shared ground antennas and monitor stations. The “master control station” (Schriever AFB) and four additional monitoring stations (on Hawaii, Ascension Islands, Diego Garcia and Kawajalein) were set up for monitoring the space segment (satellites). During August and September 2005, six more monitor stations of the NGA (National Geospatial-Intelligence Agency) were added to the grid. Now, every satellite can be seen from at least two monitor stations. This allows to calculate more precise orbits and ephemeris data. For the end user, a better position precision can be expected from this.

The user segment is composed of the passive monitor stations which are GPS receivers. GPS receivers track all satellites in their range and collect

data of the satellite signals. The raw data are then sent to the master control station where the data are processed.

Some of the sources of errors in GPS are shown in figure A.1 and explained below [2].

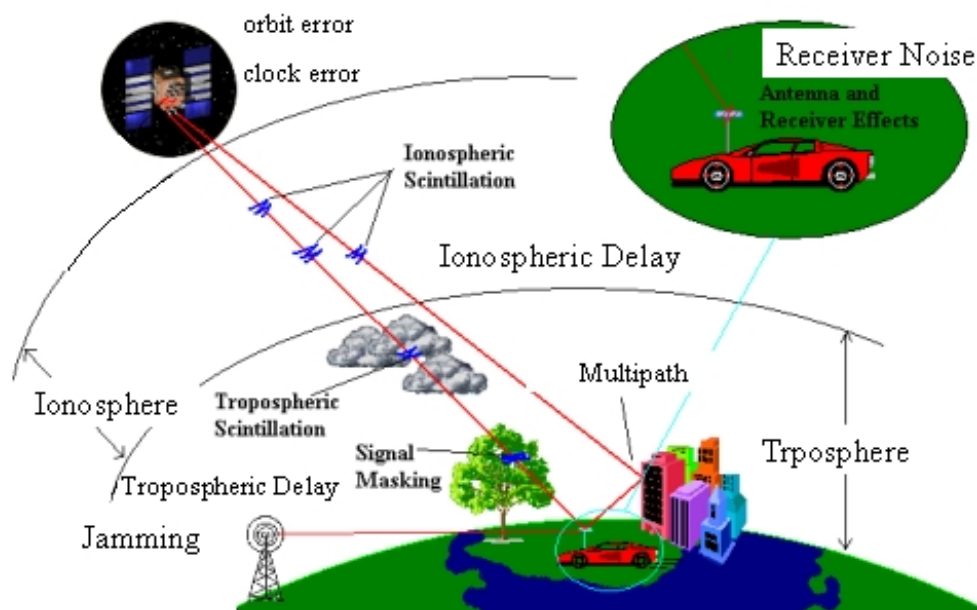


Figure A.1: Errors of GPS signal (Image courtesy of Iranmap website¹).

- **Selective availability**

Selective availability (SA), the most relevant factor for the inaccuracy of GPS system, was turned off on May 2, 2000 due to the broad use of the GPS system. SA is an artificial falsification of the time transmitted by the satellite. For civil GPS receivers that leads to a less accurate position determination (inaccuracy of 50-150 meters). The reasons for SA were safety concerns. For example terrorists should not be provided with the possibility of locating important buildings with homemade remote control weapons. Nowadays , after deactivation of SA, an accuracy of 10 - 20 meters is usual.

¹<http://www.iranmap.com/2010/04/10/gps-signal-and-errors>

- **Satellite geometry**

Satellite geometry describes the position of the satellites relative to each other from the view of the receiver. If a receiver sees 4 satellites and all are arranged for example in the north-west, this leads to a bad geometry. In the worst case, no position determination is possible at all, when all distance determinations point to the same direction. Even if a position is determined, the error of the positions may be up to 100-150 m.

- **Satellite orbits**

For satellites slight shifts of the orbits are possible due to gravitation forces. Sun and moon have a weak influence on the orbits. The orbit data are controlled and corrected regularly and are sent to the receivers. Therefore the resulting error is not more than 2 m.

- **Multipath effects**

The multipath effect is caused by reflection of satellite signals from surrounding objects, buildings, terrain, canyon walls, hard ground, etc. For GPS signals this effect mainly appears in the neighbourhood of large buildings. The reflected signal takes more time to reach the receiver than the direct signal, and these delayed signals can cause inaccuracy.

- **Atmospheric effects**

While satellite signals travel with the speed of light in the outer space, their speed is reduced in the troposphere and ionosphere. These layers refract the signals, resulting in an error.

- **Clock errors**

Despite the synchronization of the receiver clock with the satellite time during the position determination, the remaining inaccuracy of the time still leads to an error. Even a very small clock error multiplied by the very large speed of light results in a large positional error.

- **Relativistic effects**

Based on the theory of relativity, time runs slower during very fast movements. For satellites moving with a speed of 3874 m/s, clocks run slower

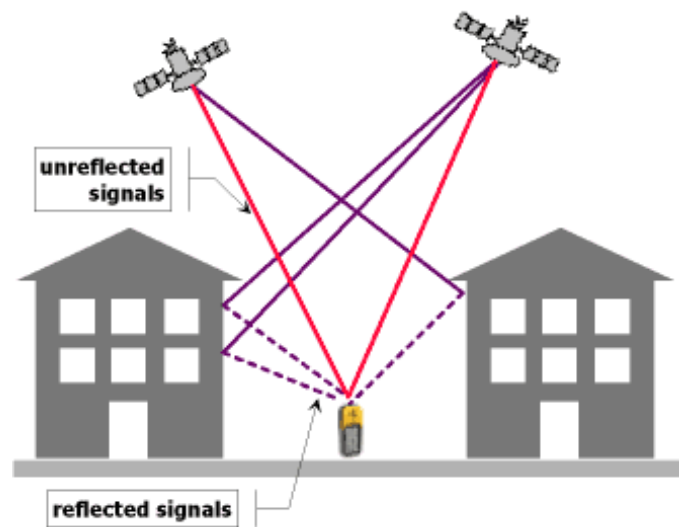


Figure A.2: Tall buildings cause the reflection of satellites' signals (Image courtesy of Kowoma website [2])

than earth. The theory of relativity also says that time moves the slower the stronger the field of gravitation is. For the satellite in 20000 km height the gravitation is weaker than earth. Altogether, the clocks of the satellites seem to run a little faster. The shift of time would be about 38 milliseconds per day and would make up for a total error of approximately 10 km per day.

Altogether the error of ± 15 meters should be expected from GPS.

Appendix B

Graph Cuts

In this section we will review the concept of graph-cuts and its application for energy minimization. As shown in Fig.B.1, an undirected graph $\mathcal{G} = \langle \nu, \varepsilon \rangle$ is defined as a set of nodes (vertices ν) and a set of undirected edges (ε) that connect these nodes. Each edge in the graph is assigned a non-negative cost. There are also two special nodes called the *source* (S) and the *sink* (T) terminals. A cut on this graph partitions the nodes into two disjoint sets, such that terminals are separated. The cost of a cut is defined as the sum of the costs of all of the severed edges. The minimum cut is the cut with the smallest cost. There are numerous algorithms for determining this minimum cut. We have used max-flow/min-cut algorithm [12] for the implementation of our stitching method.

According to the max-flow/min-cut theorem, the minimum cut can be efficiently computed by finding the maximum flow between the two terminals. The max-flow algorithm gradually increases the flow sent from the source S to the sink T along the edges in \mathcal{G} given their costs (capacities). Upon termination, the maximum flow saturates the graph. The saturated edges correspond to the minimum cost cut on \mathcal{G} , resulting in an optimal cut [11].

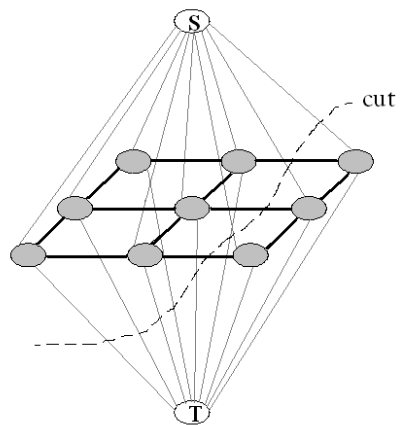


Figure B.1: A simple 2-D graph for a 3×3 image and its minimum cut.

Appendix C

Mean Standard Deviation Line Descriptor

Mean standard deviation line descriptor (MSLD) is a robust descriptor for line matching proposed by Z. Wang et al. [46]. In this section, we briefly describe this method.

Given a line segment L , as shown in Fig.C.1, two directions are introduced: line direction (d_L) and the direction of an orthogonal to the line (d_\perp). For each pixel on the line L (assuming L consists of N pixels: $P_1..P_N$), M non-overlapped sub-regions $R_1..R_M$ are defined along the direction d_\perp .

To make the line descriptor rotation invariant, each sample gradient of the image is rotated and aligned with the directions d_\perp and d_L . Then each sub-region is characterized by a descriptor vector using a SIFT-like strategy as follows: a 4-D feature vector is formed by accumulating the gradients along the directions d_\perp , d_L and their opposite directions. These vectors are shown in Fig.C.1 for two sample sub-regions:

$$\mathbf{V} = (V^1, V^2, V^3, V^4) \tag{C.1}$$

By stacking the descriptor vectors of all the sub-regions associated with a line segment, a $4M \times N$ matrix called line gradient descriptor matrix (GDM) is formed as

$$\mathbf{GDM}(L) = \begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{21} & \dots & \mathbf{V}_{N1} \\ \mathbf{V}_{12} & \mathbf{V}_{22} & \dots & \mathbf{V}_{N2} \\ \dots & \dots & \dots & \dots \\ \mathbf{V}_{1M} & \mathbf{V}_{2M} & \dots & \mathbf{V}_{NM} \end{pmatrix} \quad (\text{C.2})$$

To make the descriptor independent of the line length, mean, and standard deviation are adopted along the rows of matrix \mathbf{GDM} . And in order to make the descriptor invariant to linear changes of illumination, the mean vector and the standard deviation vector are normalized to unit norm, respectively. Final descriptor is a $8M$ -element vector obtained by concatenating these two vectors:

$$\mathbf{MSLD}(L) = \begin{pmatrix} \frac{\text{Mean}(\mathbf{GDM}(L))}{\|(\text{Mean}(\mathbf{GDM}(L)))\|} \\ \frac{\text{Std}(\mathbf{GDM}(L))}{\|(\text{Std}(\mathbf{GDM}(L)))\|} \end{pmatrix} \quad (\text{C.3})$$

In their paper, Z. Wang et al. used $M = 9$ and the size of each sub-region was 5×5 pixels [46]. We used the same values for our experiments. Therefore, our line descriptors were 72-element vectors.

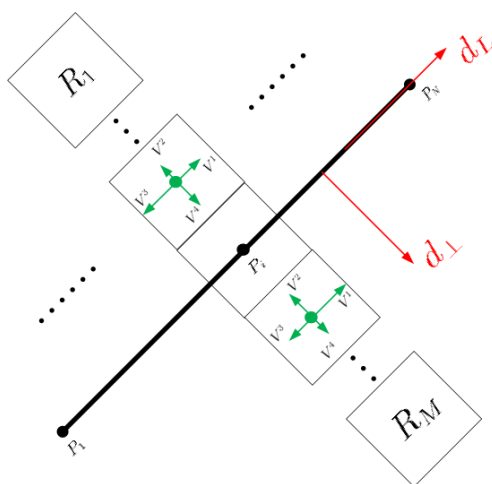


Figure C.1: Schematic figure of MSLD construction [46]

Appendix D

Chains' Information



Figure D.1: *Chain_Id* = 45614



Figure D.2: *Chain_Id* = 45616



Figure D.3: *Chain_Id* = 55764



Figure D.4: $Chain_Id = 55938$



Figure D.5: *Chain_Id* = 55964



Figure D.6: *Chain_Id* = 93849



Figure D.7: *Chain_Id* = 151323

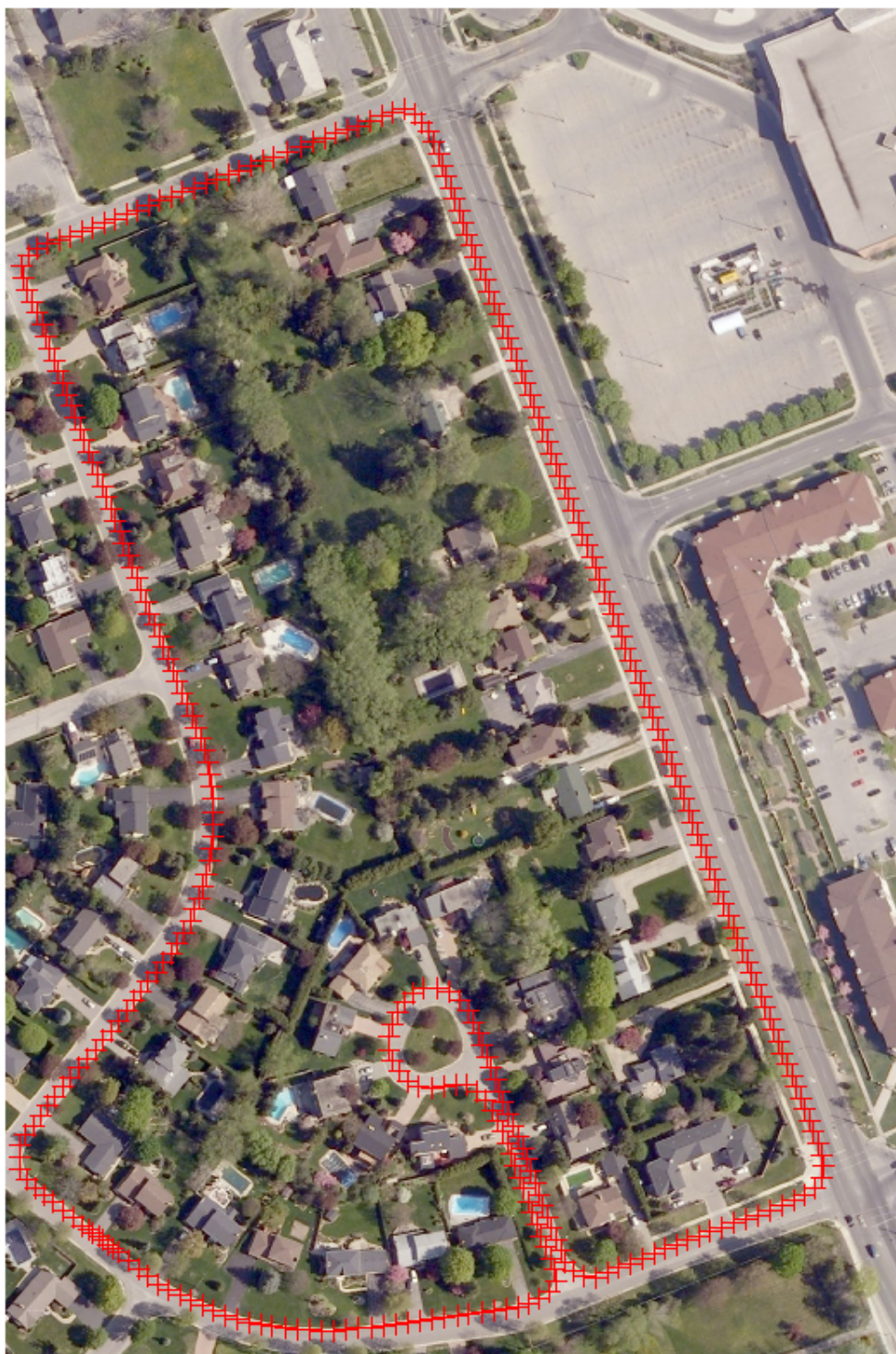


Figure D.8: *Chain_Id* = 151325

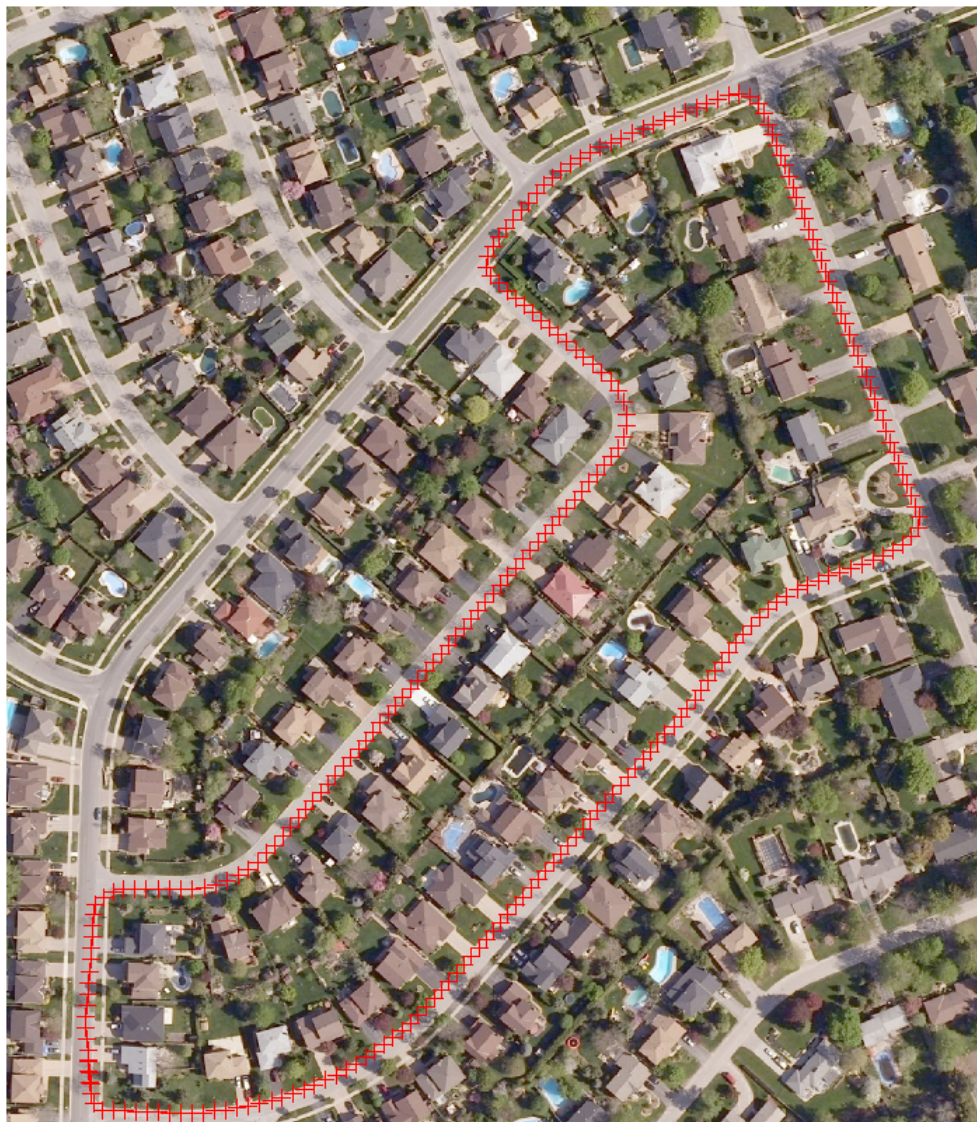


Figure D.9: *Chain_Id* = 284553



Figure D.10: *Chain_Id* = 286309

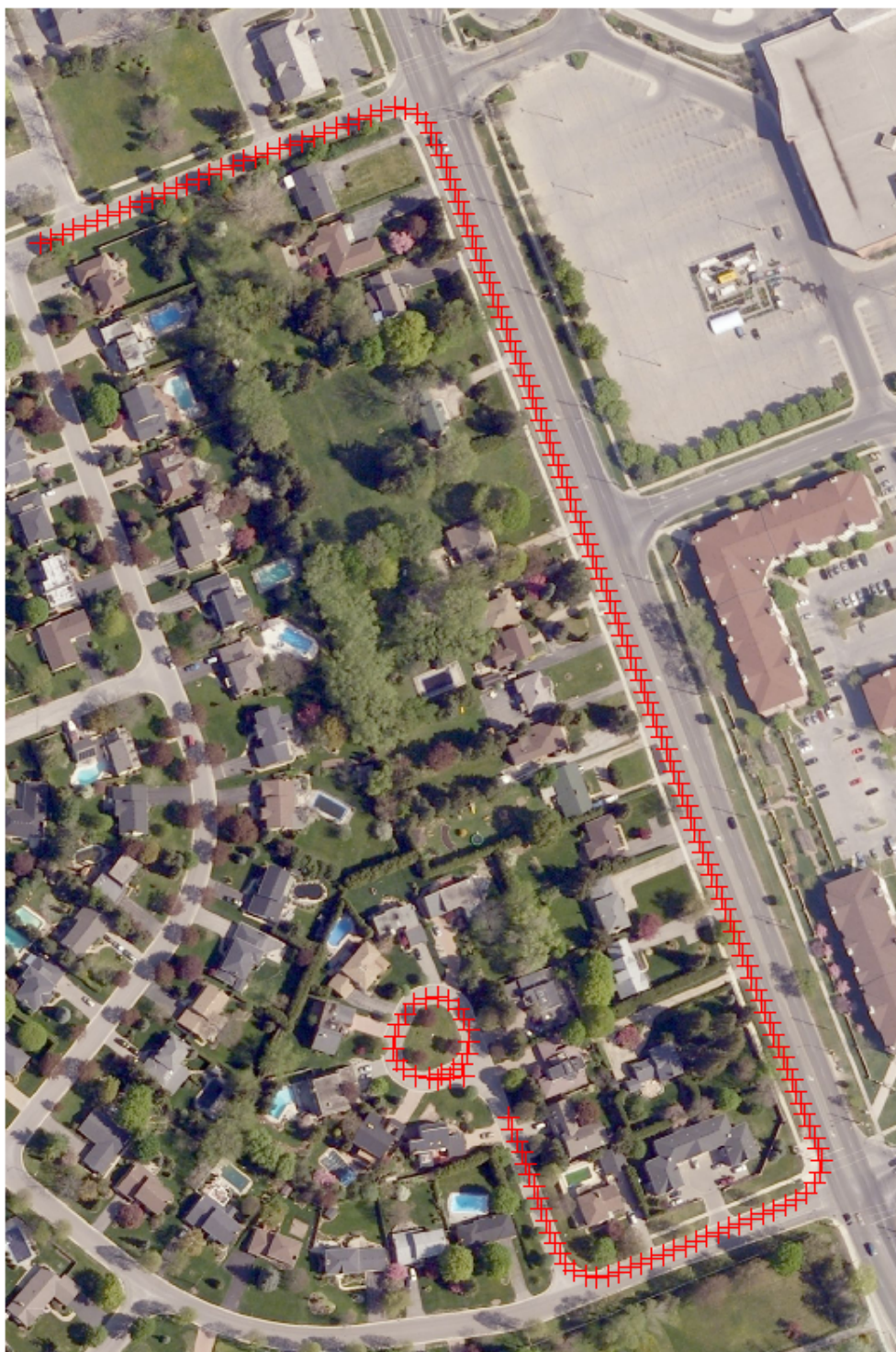


Figure D.11: *Chain_Id* = 286313

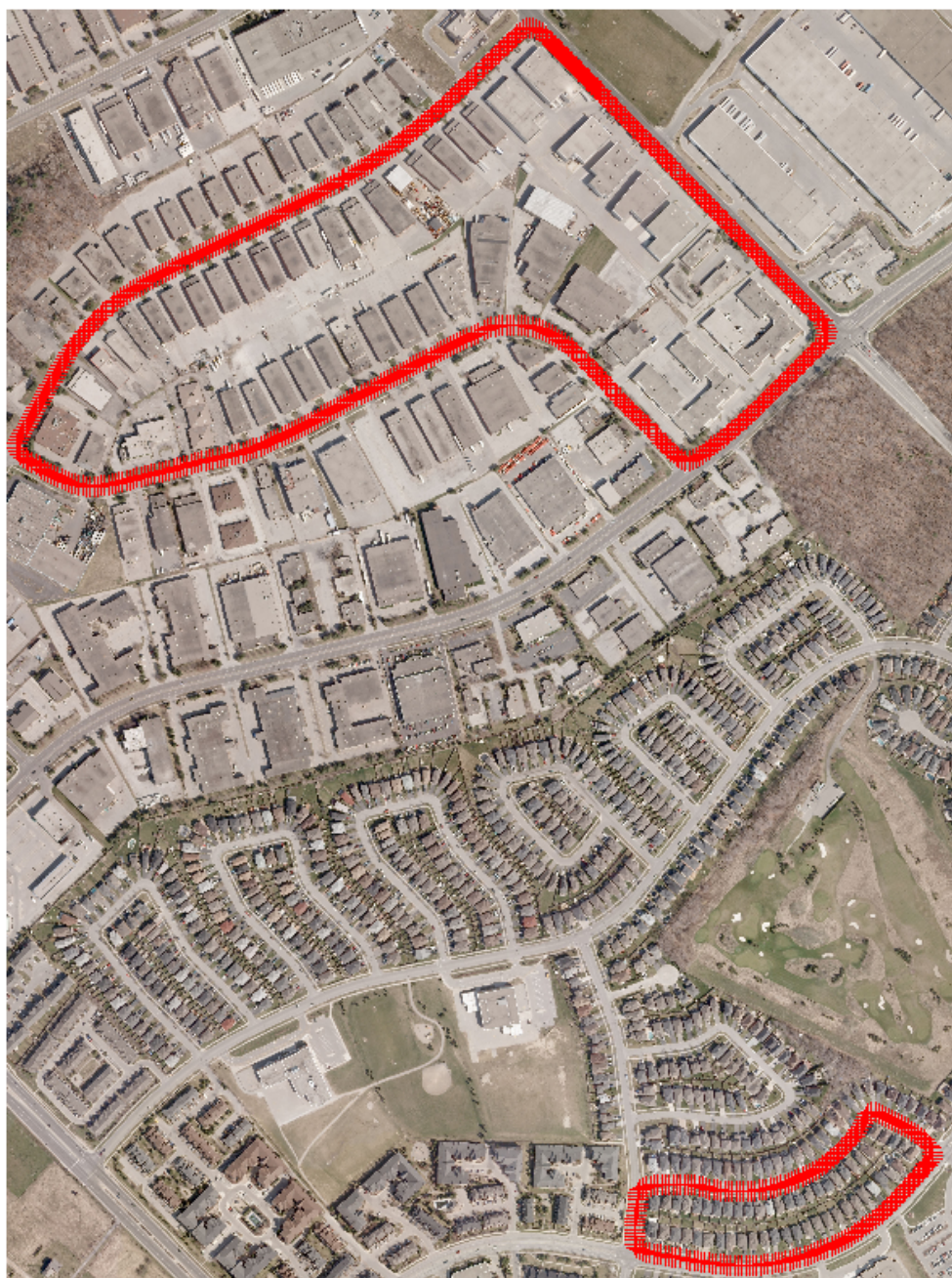


Figure D.13: *Chain_Id* = 112357



Figure D.14: *Chain_Id* = 114759



Figure D.15: *Chain_Id* = 114761



Figure D.16: *Chain_Id* = 230253



Figure D.17: *Chain_Id* = 290663



Figure D.18: *Chain_Id* = 53190

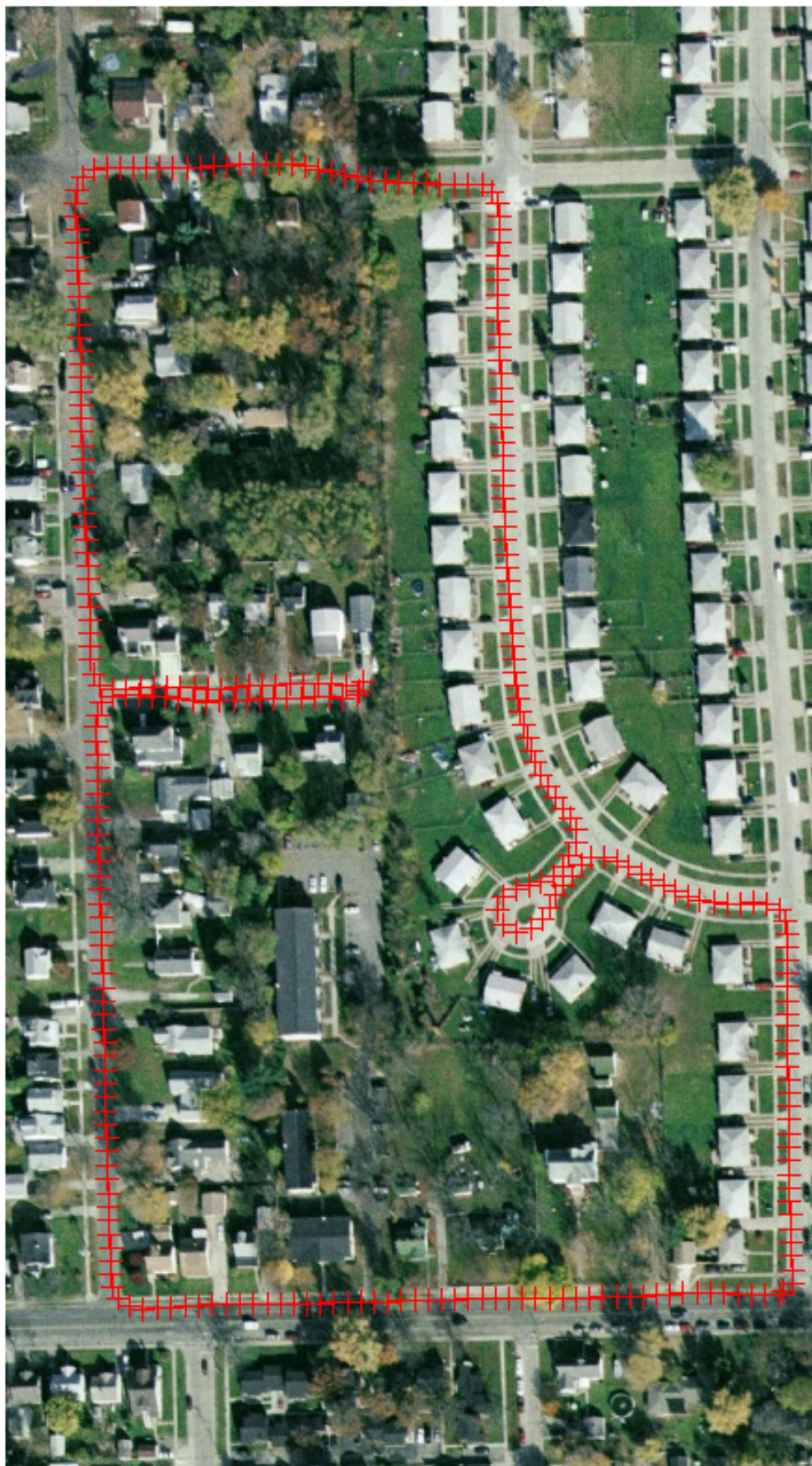


Figure D.19: *Chain.Id* = 53456



Figure D.20: *Chain_Id* = 54285

Appendix E

Results

In this appendix, the results for different stripes are presented. Each table shows the result for a different chain. In each table, the correct scale (pixel size) found by the algorithm is shown in second column. The third column indicates if the line correspondences are detected or not. The fourth column shows the score of the modified Chamfer matching. We included this score to show that MCM score is not always a reliable score for final evaluation. The fifth column shows our proposed evaluation score that should be greater than 0.2 for a pair of stripes to be accepted as the final result. The final column shows the error of the algorithm in meters.

Table E.1: Results for chainId=45614

ChainId	Scale	Line	MCM_score	eval_score	error(m)
45614	0.47	1.00	57.91	0.16	2.67
45614	0.51	1.00	44.46	0.70	1.27
45614	0.51	1.00	33.25	0.84	0.19
45614	0.51	1.00	34.58	0.84	0.92
45614	0.51	1.00	44.44	0.77	0.83
45614	0.51	1.00	41.50	0.78	0.14
45614	0.47	1.00	36.71	0.56	1.20
45614	0.51	1.00	23.95	0.31	3.10
45614	0.47	1.00	32.23	0.57	2.71
45614	0.51	1.00	34.41	0.63	3.11
45614	0.47	1.00	52.19	0.12	5.98

Table E.2: Results for chainId=45616

ChainId	Scale	Line	MCM_score	eval_score	error(m)
45616	0.47	1.00	30.03	0.56	2.60
45616	0.51	1.00	44.09	0.45	2.86
45616	0.51	1.00	39.09	0.20	3.46
45616	0.51	1.00	41.73	0.15	4.07
45616	0.49	1.00	37.64	0.38	0.75
45616	0.49	1.00	36.68	0.52	1.61
45616	0.49	1.00	50.88	0.49	2.94
45616	0.49	1.00	47.47	0.06	7.10
45616	0.49	1.00	117.67	0.01	9.04
45616	0.49	1.00	35.67	0.24	2.00
45616	0.47	1.00	73.06	0.08	3.72

Table E.3: Results for chainId=55764

ChainId	Scale	Line	MCM_score	eval_score	error(m)
55764	0.47	1.00	51.00	0.73	1.47
55764	0.51	1.00	72.85	0.51	2.42
55764	0.49	1.00	52.16	0.53	2.23
55764	0.51	1.00	96.25	0.24	2.49
55764	0.47	1.00	47.84	0.26	1.43
55764	0.51	1.00	38.90	0.64	2.08
55764	0.51	1.00	65.07	0.70	1.36
55764	0.47	1.00	69.32	0.54	1.31

Table E.4: Results for chainId=55938

ChainId	Scale	Line	MCM_score	eval_score	error(m)
55938	0.47	0.00	61.82	0.15	4.59
55938	0.47	1.00	57.23	0.06	7.01
55938	0.49	0.00	50.80	0.24	6.58
55938	0.51	1.00	37.79	0.19	5.95
55938	0.49	0.00	43.36	0.17	5.70
55938	0.47	1.00	67.05	0.29	2.05
55938	0.51	1.00	39.97	0.45	3.12
55938	0.51	1.00	43.20	0.42	3.05
55938	0.47	1.00	34.17	0.40	2.81
55938	0.49	1.00	44.98	0.24	2.08
55938	0.47	1.00	97.67	0.14	5.30
55938	0.49	0.00	30.43	0.04	3.98
55938	0.51	0.00	40.29	0.14	3.48
55938	0.51	0.00	39.28	0.14	4.13
55938	0.51	1.00	34.62	0.30	0.34
55938	0.47	1.00	48.29	0.38	0.64
55938	0.51	1.00	48.74	0.47	0.87
55938	0.49	1.00	52.15	0.27	5.17
55938	0.51	1.00	46.15	0.03	3.17
55938	0.47	1.00	40.12	0.15	6.44
55938	0.51	1.00	33.17	0.18	7.30
55938	0.51	1.00	41.02	0.35	1.16
55938	0.49	1.00	37.84	0.34	1.90
55938	0.51	1.00	35.89	0.38	1.87
55938	0.47	0.00	43.52	0.35	0.29
55938	0.47	0.00	56.56	0.44	0.18
55938	0.51	0.00	59.03	0.02	3.97
55938	0.47	1.00	76.78	0.31	6.75
55938	0.47	1.00	55.28	0.38	2.19
55938	0.51	1.00	54.44	0.27	4.39
55938	0.51	1.00	50.17	0.31	1.30
55938	0.51	1.00	47.02	0.34	1.22
55938	0.49	1.00	47.03	0.44	0.09
55938	0.51	1.00	36.00	0.60	1.14
55938	0.49	1.00	35.51	0.49	0.74
55938	0.47	0.00	45.42	0.23	2.20
55938	0.51	1.00	49.88	0.24	1.74
55938	0.49	1.00	42.91	0.22	2.16
55938	0.47	1.00	50.96	0.21	4.70
55938	0.49	1.00	46.00	0.04	7.11
55938	0.47	1.00	38.40	0.41	4.73
55938	0.51	1.00	37.77	0.28	3.02
To be	continued	in	the	Next	Page

Table E.5: Results for chainId=55938 (continue)

ChainId	Scale	Line	MCM_score	eval_score	error(m)
55938	0.47	0.00	41.71	0.18	1.45
55938	0.47	1.00	51.30	0.54	2.07
55938	0.49	1.00	39.47	0.46	2.34
55938	0.47	1.00	36.66	0.44	2.07
55938	0.51	1.00	34.80	0.72	0.90
55938	0.49	1.00	36.83	0.47	0.08
55938	0.51	1.00	36.78	0.53	1.56
55938	0.51	1.00	41.26	0.03	4.50
55938	0.51	1.00	48.54	0.21	5.91
55938	0.47	0.00	41.95	0.29	2.95
55938	0.51	1.00	56.55	0.32	1.06
55938	0.51	1.00	56.54	0.17	4.28
55938	0.51	1.00	42.82	0.52	1.86
55938	0.51	1.00	44.48	0.55	0.62
55938	0.51	1.00	42.26	0.43	0.60
55938	0.51	0.00	39.98	0.46	1.16
55938	0.51	1.00	45.62	0.37	0.29
55938	0.51	1.00	43.44	0.46	1.98
55938	0.51	1.00	41.66	0.48	2.38
55938	0.47	0.00	48.53	0.11	0.65
55938	0.47	1.00	98.06	0.73	3.45
55938	0.47	1.00	111.08	0.63	0.34
55938	0.51	0.00	40.90	0.36	2.51

Table E.6: Results for chainId=55964

ChainId	Scale	Line	MCM_score	eval_score	error(m)
55964	0.47	1.00	41.64	0.52	0.57
55964	0.47	1.00	38.52	0.57	0.05
55964	0.51	1.00	35.57	0.60	0.58
55964	0.47	1.00	33.54	0.14	0.63
55964	0.51	1.00	57.96	0.75	0.56
55964	0.49	1.00	48.07	0.56	2.20
55964	0.51	1.00	36.51	0.63	1.12
55964	0.51	1.00	37.24	0.72	0.21
55964	0.49	1.00	25.32	0.63	1.70
55964	0.49	1.00	29.52	0.48	1.93
55964	0.51	1.00	33.94	0.45	1.47
55964	0.49	1.00	39.66	0.30	1.47
55964	0.51	1.00	33.63	0.39	0.75
55964	0.47	1.00	38.04	0.56	1.83
55964	0.51	0.00	32.39	0.43	0.92
55964	0.49	1.00	32.52	0.54	1.04
55964	0.49	1.00	35.79	0.36	2.08
55964	0.51	0.00	30.25	0.04	1.34
55964	0.47	0.00	47.81	0.03	4.35
55964	0.51	1.00	177.24	0.14	13.00
55964	0.47	1.00	25.41	0.36	4.69
55964	0.49	1.00	30.42	0.41	4.58
55964	0.47	1.00	42.95	0.47	3.68
55964	0.47	1.00	49.25	0.32	3.93
55964	0.51	1.00	37.99	0.47	3.14
55964	0.51	1.00	37.03	0.53	10.75
55964	0.51	1.00	80.80	0.46	3.12
55964	0.49	1.00	31.95	0.51	0.95

Table E.7: Results for chainId=93849

ChainId	Scale	Line	MCM_score	eval_score	error(m)
93849	0.51	1.00	57.74	0.07	2.46
93849	0.49	1.00	51.37	0.23	1.77
93849	0.47	1.00	53.89	0.14	1.35
93849	0.51	1.00	54.11	0.29	0.63
93849	0.47	0.00	50.72	0.06	2.78
93849	0.47	1.00	82.10	0.53	3.30
93849	0.51	1.00	87.62	0.67	0.57
93849	0.47	1.00	66.32	0.17	2.50
93849	0.47	1.00	83.20	0.31	3.32
93849	0.51	1.00	45.14	0.19	1.11
93849	0.51	1.00	56.12	0.32	1.35

Table E.8: Results for chainId=151323

ChainId	Scale	Line	MCM_score	eval_score	error(m)
151323	0.51	1.00	27.61	0.91	3.05
151323	0.51	1.00	64.15	0.19	6.92
151323	0.47	1.00	54.22	0.61	3.10
151323	0.47	1.00	52.45	0.25	2.57
151323	0.49	1.00	48.97	0.61	1.81
151323	0.47	1.00	40.46	0.58	2.33
151323	0.51	1.00	36.59	0.74	2.68
151323	0.49	1.00	61.17	0.59	4.14
151323	0.47	1.00	50.87	0.69	1.86
151323	0.51	1.00	43.69	0.89	2.16
151323	0.47	1.00	37.72	0.70	1.81
151323	0.49	1.00	31.17	0.69	2.25
151323	0.51	1.00	27.74	0.70	3.01
151323	0.51	1.00	51.51	0.52	1.26
151323	0.49	0.00	35.78	0.34	2.84
151323	0.47	0.00	31.99	0.34	3.35
151323	0.49	0.00	37.69	0.47	0.87
151323	0.51	0.00	40.81	0.18	2.37
151323	0.47	0.00	37.98	0.07	3.06
151323	0.51	0.00	37.00	0.32	1.64

Table E.9: Results for chainId=151325

ChainId	Scale	Line	MCM.score	eval.score	error(m)
151325	0.49	1.00	116.94	0.30	3.39
151325	0.51	1.00	64.72	0.47	2.10
151325	0.47	1.00	70.68	0.03	3.03
151325	0.47	1.00	108.71	0.35	0.12
151325	0.51	1.00	104.36	0.33	2.69
151325	0.49	1.00	50.29	0.49	0.17
151325	0.51	1.00	57.38	0.46	2.62
151325	0.49	1.00	205.07	0.25	0.03
151325	0.51	1.00	90.27	0.47	1.03
151325	0.47	1.00	75.80	0.09	2.33
151325	0.47	1.00	73.26	0.14	3.07
151325	0.49	0.00	98.92	0.04	3.45
151325	0.47	0.00	56.46	0.58	3.07
151325	0.47	0.00	40.82	0.50	2.32
151325	0.49	0.00	40.41	0.19	3.56
151325	0.51	0.00	31.27	0.04	5.06
151325	0.49	0.00	27.29	0.16	7.14
151325	0.49	0.00	39.67	0.12	3.82
151325	0.47	1.00	35.09	0.84	1.39
151325	0.47	0.00	25.88	0.16	7.03
151325	0.49	0.00	27.59	0.07	3.93
151325	0.51	0.00	32.91	0.00	1.90
151325	0.51	0.00	33.57	0.28	3.29
151325	0.47	1.00	37.52	0.46	1.84
151325	0.47	0.00	57.76	0.05	6.48
151325	0.47	0.00	38.80	0.17	2.82
151325	0.51	0.00	40.19	0.09	3.37
151325	0.51	1.00	49.40	0.19	4.30
151325	0.49	0.00	41.14	0.05	3.82
151325	0.47	0.00	38.39	0.19	5.52
151325	0.49	0.00	39.72	0.00	3.08
151325	0.49	1.00	29.04	0.58	1.94
151325	0.47	1.00	32.13	0.59	3.18
151325	0.47	1.00	49.87	0.53	2.81
151325	0.51	1.00	73.55	0.07	2.66
151325	0.51	0.00	90.50	0.13	6.09

Table E.10: Results for chainId=284553

ChainId	Scale	Line	MCM_score	eval_score	error(m)
284553	0.51	0.00	45.67	0.33	1.50
284553	0.47	1.00	55.49	0.24	0.43
284553	0.47	1.00	41.83	0.05	1.60
284553	0.47	1.00	44.18	0.34	1.36
284553	0.49	1.00	40.86	0.26	1.72
284553	0.51	1.00	57.73	0.43	2.99
284553	0.47	1.00	38.64	0.19	1.95
284553	0.47	1.00	53.15	0.14	4.31
284553	0.47	1.00	52.98	0.04	3.12
284553	0.51	1.00	54.46	0.08	1.91
284553	0.47	1.00	90.10	0.31	4.01
284553	0.51	1.00	49.66	0.00	0.73
284553	0.47	1.00	45.16	0.08	0.07
284553	0.51	1.00	66.53	0.18	2.60
284553	0.49	1.00	48.76	0.08	0.37
284553	0.51	0.00	106.57	0.22	2.95
284553	0.51	1.00	58.85	0.42	1.53
284553	0.51	1.00	59.44	0.59	1.59
284553	0.49	1.00	109.26	0.09	3.09
284553	0.49	1.00	79.41	0.01	11.72
284553	0.47	1.00	68.76	0.13	4.52
284553	0.51	1.00	59.96	0.02	7.43
284553	0.49	1.00	39.97	0.67	0.72
284553	0.51	1.00	97.83	0.62	0.52
284553	0.47	1.00	87.29	0.41	1.18
284553	0.47	0.00	66.76	0.24	3.77
284553	0.51	1.00	53.82	0.10	11.33
284553	0.51	1.00	44.25	0.66	1.00
284553	0.51	1.00	43.76	0.67	1.26
284553	0.51	1.00	29.07	0.74	1.62
284553	0.51	0.00	57.19	0.38	4.96
284553	0.51	1.00	87.45	0.57	0.53
284553	0.47	1.00	42.06	0.23	3.85
284553	0.51	1.00	87.91	0.02	6.02
284553	0.47	1.00	73.33	0.64	1.10
284553	0.51	1.00	84.68	0.74	1.19
284553	0.47	1.00	79.96	0.16	2.31
284553	0.51	1.00	106.84	0.31	3.40
284553	0.51	1.00	135.28	0.05	1.67

Table E.11: Results for chainId=286309

ChainId	Scale	Line	MCM_score	eval_score	error(m)
286309	0.47	1.00	60.90	0.16	5.00
286309	0.49	1.00	99.44	0.01	11.79
286309	0.51	1.00	48.16	0.02	5.35
286309	0.47	1.00	102.27	0.32	0.47
286309	0.49	1.00	64.30	0.08	5.63
286309	0.51	1.00	72.91	0.06	5.56
286309	0.47	1.00	85.23	0.07	5.18
286309	0.47	1.00	37.30	0.34	1.69
286309	0.47	1.00	36.24	0.29	1.79
286309	0.49	1.00	86.86	0.33	2.76
286309	0.47	1.00	52.91	0.70	1.35
286309	0.51	1.00	52.06	0.84	1.46
286309	0.51	1.00	31.80	0.82	0.85
286309	0.49	1.00	39.81	0.35	1.49
286309	0.47	0.00	47.63	0.10	2.64
286309	0.49	1.00	52.56	0.10	4.26
286309	0.51	1.00	56.47	0.24	3.18
286309	0.47	1.00	71.34	0.27	3.82
286309	0.49	1.00	71.76	0.18	4.09
286309	0.47	1.00	44.70	0.37	2.89
286309	0.47	1.00	58.98	0.04	3.43
286309	0.49	1.00	58.82	0.48	2.12
286309	0.47	1.00	45.88	0.55	1.51
286309	0.49	1.00	33.48	0.43	0.38
286309	0.47	1.00	36.43	0.40	1.60
286309	0.47	0.00	62.72	0.27	3.26
286309	0.49	1.00	90.98	0.12	1.04
286309	0.51	1.00	82.72	0.42	4.55
286309	0.51	1.00	90.19	0.16	3.89
286309	0.51	1.00	66.92	0.60	0.63
286309	0.51	1.00	112.19	0.12	1.95
286309	0.51	0.00	38.93	0.35	5.56
286309	0.51	0.00	71.98	0.27	4.09
286309	0.49	0.00	59.62	0.10	3.85
286309	0.47	0.00	43.63	0.16	6.33
286309	0.49	0.00	43.21	0.12	6.00
286309	0.47	1.00	105.37	0.48	0.96
286309	0.47	0.00	80.49	0.23	3.37

Table E.12: Results for chainId=286313

ChainId	Scale	Line	MCM_score	eval_score	error(m)
286313	0.47	1.00	42.73	0.42	2.82
286313	0.49	1.00	55.62	0.33	2.01
286313	0.49	1.00	57.59	0.26	1.92
286313	0.51	1.00	33.05	0.33	5.18
286313	0.51	1.00	53.08	0.19	0.57
286313	0.51	1.00	56.00	0.25	0.43
286313	0.51	1.00	41.50	0.31	0.31
286313	0.47	1.00	32.33	0.56	1.06
286313	0.51	1.00	33.22	0.46	1.81
286313	0.47	1.00	56.29	0.09	2.53
286313	0.47	0.00	46.06	0.08	2.82

Table E.13: Results for chainId=287607

ChainId	Scale	Line	MCM_score	eval_score	error(m)
287607	0.49	1.00	71.92	0.10	1.33
287607	0.51	1.00	54.65	0.06	2.06
287607	0.51	1.00	49.55	0.39	2.34
287607	0.51	1.00	50.74	0.81	3.21
287607	0.47	1.00	56.21	0.13	1.71
287607	0.51	0.00	72.35	0.14	5.41
287607	0.47	0.00	47.51	0.18	3.22
287607	0.51	0.00	43.48	0.09	2.80
287607	0.51	0.00	53.22	0.04	1.41
287607	0.51	0.00	46.87	0.13	2.45
287607	0.47	1.00	52.51	0.42	1.26
287607	0.51	1.00	85.69	0.08	9.79
287607	0.47	1.00	75.95	0.05	7.73

Table E.14: Results for chainId=112357

ChainId	Scale	Line	MCM_score	eval_score	error(m)
112357	0.51	1.00	25.99	0.61	5.53
112357	0.51	1.00	29.83	0.47	1.07
112357	0.47	1.00	25.38	0.40	1.45
112357	0.47	0.00	43.47	0.74	0.10
112357	0.51	1.00	33.97	0.80	0.45
112357	0.51	1.00	30.88	0.80	0.95
112357	0.51	1.00	43.78	0.25	3.70
112357	0.51	1.00	52.58	0.26	3.72
112357	0.47	0.00	38.55	0.60	0.41
112357	0.47	1.00	78.86	0.56	0.32
112357	0.47	1.00	40.67	0.08	4.88
112357	0.47	1.00	40.45	0.05	0.62
112357	0.51	1.00	22.65	0.57	2.95
112357	0.51	1.00	67.96	0.40	0.99
112357	0.51	1.00	60.55	0.41	1.40
112357	0.51	1.00	67.16	0.34	1.98
112357	0.51	1.00	67.74	0.61	1.10
112357	0.51	1.00	91.42	0.38	1.07
112357	0.47	1.00	107.11	0.42	0.26
112357	0.51	1.00	82.69	0.56	0.94
112357	0.49	1.00	47.70	0.31	0.99
112357	0.51	1.00	71.75	0.29	1.53
112357	0.47	1.00	76.99	0.59	1.48
112357	0.51	1.00	41.66	0.33	2.02
112357	0.47	1.00	83.72	0.42	0.54
112357	0.47	1.00	54.50	0.58	0.30
112357	0.51	1.00	70.00	0.54	0.43
112357	0.51	1.00	67.08	0.48	0.96
112357	0.49	1.00	77.50	0.43	0.89
112357	0.51	1.00	77.07	0.35	0.80
112357	0.51	1.00	61.82	0.44	1.44
112357	0.47	1.00	72.02	0.41	0.71
112357	0.51	1.00	53.88	0.38	0.92
112357	0.47	1.00	49.41	0.04	3.50
112357	0.47	1.00	115.52	0.27	4.19
112357	0.51	0.00	89.14	0.13	3.96
112357	0.51	0.00	94.07	0.12	1.16
112357	0.47	1.00	46.99	0.00	4.06
112357	0.47	1.00	96.12	0.06	4.11
112357	0.51	1.00	67.62	0.40	1.07
112357	0.51	1.00	77.24	0.27	2.24
112357	0.51	1.00	63.41	0.41	1.77
To be	continued	in	the	Next	Page

Table E.15: Results for chainId=112357 (continue)

ChainId	Scale	Line	MCM_score	eval_score	error(m)
112357	0.49	1.00	78.75	0.33	0.35
112357	0.47	1.00	71.51	0.30	0.04
112357	0.51	1.00	65.46	0.17	3.20
112357	0.51	1.00	56.64	0.40	0.86
112357	0.51	1.00	61.85	0.48	1.17
112357	0.51	0.00	66.09	0.40	1.02
112357	0.47	1.00	64.56	0.31	0.24
112357	0.51	0.00	47.38	0.06	1.74
112357	0.49	1.00	74.71	0.33	0.76
112357	0.47	1.00	70.28	0.27	0.76
112357	0.47	1.00	69.05	0.02	0.67
112357	0.49	1.00	63.38	0.25	3.01
112357	0.51	1.00	67.31	0.28	5.04
112357	0.51	1.00	34.82	0.60	0.46
112357	0.49	1.00	35.90	0.74	0.23
112357	0.51	1.00	44.68	0.83	0.60
112357	0.47	1.00	49.52	0.36	0.97
112357	0.51	1.00	42.42	0.63	1.33
112357	0.51	1.00	54.50	0.35	2.98
112357	0.47	1.00	80.25	0.09	0.61

Table E.16: Results for chainId=114759

ChainId	Scale	Line	MCM_score	eval_score	error(m)
114759	0.51	1.00	55.94	0.37	1.03
114759	0.47	1.00	36.11	0.44	1.45
114759	0.49	1.00	39.68	0.48	1.07
114759	0.51	1.00	62.37	0.54	0.78
114759	0.51	1.00	31.23	0.61	1.04
114759	0.51	1.00	33.11	0.53	1.58
114759	0.49	1.00	20.21	0.44	1.49
114759	0.51	0.00	49.55	0.22	3.93
114759	0.51	0.00	52.60	0.50	1.42
114759	0.49	1.00	25.12	0.20	1.05
114759	0.47	1.00	38.90	0.36	1.95
114759	0.51	1.00	27.24	0.59	1.66
114759	0.49	1.00	38.35	0.63	0.88
114759	0.49	1.00	47.47	0.59	0.65
114759	0.51	1.00	39.52	0.70	0.58
114759	0.51	1.00	96.66	0.46	1.64
114759	0.47	1.00	39.53	0.16	3.49
114759	0.49	1.00	62.07	0.58	1.06
114759	0.47	1.00	44.68	0.53	0.83
114759	0.51	1.00	46.48	0.43	1.83
114759	0.51	1.00	51.05	0.43	1.67
114759	0.51	1.00	62.89	0.25	3.48
114759	0.51	1.00	60.82	0.15	4.10
114759	0.47	1.00	32.88	0.28	2.15
114759	0.47	1.00	28.91	0.57	2.05
114759	0.49	1.00	30.41	0.60	2.40
114759	0.49	1.00	60.03	0.58	2.67
114759	0.47	1.00	49.39	0.10	0.64
114759	0.51	1.00	111.13	0.12	2.10
114759	0.47	1.00	36.50	0.43	1.99
114759	0.51	1.00	29.33	0.61	0.05
114759	0.47	1.00	38.09	0.25	0.72
114759	0.51	1.00	45.90	0.47	0.46
114759	0.51	1.00	42.88	0.71	1.10
114759	0.47	1.00	72.84	0.56	1.22
114759	0.51	1.00	45.71	0.29	2.59
114759	0.51	1.00	38.26	0.38	2.18
114759	0.51	1.00	38.65	0.16	2.33
114759	0.49	0.00	32.23	0.26	2.36
114759	0.51	1.00	44.93	0.43	2.38
114759	0.47	1.00	54.44	0.06	3.20
To be	continued	in	the	Next	Page

Table E.17: Results for chainId=114759 (continue)

ChainId	Scale	Line	MCM_score	eval_score	error(m)
114759	0.51	1.00	45.97	0.19	3.37
114759	0.47	1.00	38.70	0.52	1.13
114759	0.51	1.00	40.09	0.58	1.46
114759	0.49	1.00	52.62	0.51	2.46
114759	0.51	1.00	36.94	0.35	1.84
114759	0.49	1.00	40.55	0.53	0.76
114759	0.49	1.00	45.64	0.24	6.11
114759	0.51	1.00	40.58	0.40	0.25
114759	0.49	1.00	42.30	0.49	1.46
114759	0.51	1.00	36.04	0.57	2.18
114759	0.47	1.00	29.81	0.12	6.05
114759	0.51	1.00	31.85	0.00	8.39
114759	0.51	1.00	42.35	0.40	0.65
114759	0.51	1.00	42.16	0.30	1.55
114759	0.49	1.00	37.72	0.40	0.61
114759	0.51	1.00	40.71	0.56	0.86
114759	0.51	1.00	40.97	0.43	0.98
114759	0.51	1.00	38.66	0.68	1.41
114759	0.51	1.00	31.93	0.60	2.21
114759	0.51	1.00	28.10	0.60	1.84
114759	0.51	1.00	45.17	0.59	1.51
114759	0.49	1.00	51.47	0.13	2.14
114759	0.47	1.00	33.59	0.66	0.31

Table E.18: Results for chainId=114761

ChainId	Scale	Line	MCM_score	eval_score	error(m)
114761	0.51	1.00	37.32	0.39	0.50
114761	0.51	1.00	40.22	0.17	0.95
114761	0.51	1.00	34.70	0.41	0.50
114761	0.49	1.00	46.20	0.61	1.03
114761	0.49	1.00	48.61	0.38	1.17
114761	0.51	1.00	35.61	0.02	0.41
114761	0.49	1.00	38.69	0.04	0.45
114761	0.47	0.00	45.22	0.42	1.22
114761	0.51	1.00	28.69	0.51	1.40
114761	0.51	0.00	24.99	0.32	1.51
114761	0.49	0.00	29.07	0.46	0.78
114761	0.49	0.00	34.96	0.35	0.66
114761	0.51	0.00	26.55	0.34	1.30
114761	0.49	0.00	27.97	0.19	1.72
114761	0.47	0.00	24.22	0.02	1.34
114761	0.47	0.00	46.51	0.02	1.02
114761	0.51	1.00	45.04	0.40	0.93
114761	0.51	0.00	30.06	0.29	1.77
114761	0.47	1.00	35.26	0.16	1.89
114761	0.51	1.00	43.54	0.15	2.14

Table E.19: Results for chainId=230253

ChainId	Scale	Line	MCM_score	eval_score	error(m)
230253	0.49	0.00	59.50	0.53	1.50
230253	0.47	1.00	79.94	0.51	2.20
230253	0.51	1.00	51.87	0.43	2.38
230253	0.51	0.00	87.26	0.10	2.55
230253	0.47	1.00	55.21	0.04	4.77
230253	0.51	1.00	74.20	0.24	2.69

Table E.20: Results for chainId=290663

ChainId	Scale	Line	MCM_score	eval_score	error(m)
290663	0.51	0.00	30.77	0.08	10.22
290663	0.47	0.00	43.66	0.14	6.32
290663	0.51	0.00	47.93	0.20	4.54
290663	0.47	0.00	47.11	0.02	4.79
290663	0.49	1.00	54.54	0.65	1.56
290663	0.47	1.00	89.72	0.41	0.66
290663	0.49	1.00	57.82	0.49	1.00
290663	0.51	1.00	49.14	0.48	1.13
290663	0.47	1.00	45.31	0.05	0.03
290663	0.51	1.00	45.18	0.18	2.22
290663	0.47	1.00	47.33	0.20	1.19
290663	0.51	0.00	188.65	0.02	5.55

Table E.21: Results for chainId=53190

ChainId	Scale	Line	MCM_score	eval_score	error(m)
53190	0.49	1.00	58.49	0.30	3.62
53190	0.47	0.00	55.92	0.04	5.96
53190	0.49	1.00	44.25	0.57	3.34
53190	0.51	1.00	64.56	0.50	3.45
53190	0.51	1.00	43.07	0.15	9.26
53190	0.47	1.00	54.18	0.63	2.23
53190	0.47	1.00	34.57	0.31	1.05
53190	0.47	1.00	41.40	0.34	0.59
53190	0.51	1.00	39.72	0.59	1.58
53190	0.47	1.00	68.28	0.19	5.79
53190	0.51	1.00	78.64	0.11	5.87
53190	0.51	1.00	37.63	0.46	1.11
53190	0.47	1.00	135.80	0.21	2.00
53190	0.47	1.00	121.42	0.33	0.58
53190	0.49	1.00	52.75	0.10	4.38
53190	0.51	1.00	99.63	0.41	1.81
53190	0.51	1.00	63.83	0.01	7.54
53190	0.51	1.00	75.88	0.11	1.01
53190	0.47	1.00	107.01	0.05	2.13
53190	0.51	1.00	95.19	0.11	3.31
53190	0.49	1.00	59.86	0.04	1.32
53190	0.51	1.00	89.32	0.45	3.82
53190	0.51	1.00	71.74	0.40	2.94

Table E.22: Results for chainId=53456

ChainId	Scale	Line	MCM_score	eval_score	error(m)
53456	0.51	1.00	40.57	0.43	4.15
53456	0.47	1.00	28.03	0.47	0.97
53456	0.51	1.00	27.79	0.47	0.48
53456	0.51	0.00	51.29	0.13	0.61
53456	0.47	0.00	73.34	0.02	3.98
53456	0.51	0.00	57.47	0.44	0.40
53456	0.49	1.00	58.84	0.07	1.60
53456	0.47	0.00	40.22	0.18	4.72
53456	0.47	0.00	37.11	0.16	3.27
53456	0.51	0.00	91.61	0.21	2.35
53456	0.51	0.00	100.94	0.28	6.07
53456	0.47	1.00	88.85	0.15	6.50
53456	0.51	0.00	53.96	0.06	3.04
53456	0.47	0.00	63.59	0.12	6.91
53456	0.51	1.00	35.43	0.42	1.38
53456	0.47	1.00	34.73	0.48	1.63
53456	0.49	1.00	32.43	0.55	2.34

Table E.23: Results for chainId=54285

ChainId	Scale	Line	MCM_score	eval_score	error(m)
54285	0.47	0.00	36.59	0.07	9.76
54285	0.51	1.00	91.06	0.09	9.34
54285	0.51	1.00	89.51	0.10	3.75
54285	0.47	0.00	71.71	0.07	3.09
54285	0.49	1.00	56.60	0.64	1.32
54285	0.51	1.00	37.36	0.51	1.95
54285	0.51	0.00	42.14	0.31	4.23

Bibliography

- [1] GIS solutions for urban and regional planning. <http://www.esri.com/library/brochures/pdfs/gis-sols-for-urban-planning.pdf>.
- [2] The GPS system. <http://www.kowoma.de/en/gps/satellites.htm>.
- [3] Mutual information for image registration and feature selection. http://www.cse.msu.edu/%7Ecse902/S03/mut_info.ppt.
- [4] Visualizing the real world. <http://www.atcity.com/>.
- [5] What is GIS? <http://www.esri.com>.
- [6] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver. Google street view: capturing the world at street level. *IEEE trans. on Computer*, pages 32–38, 2010.
- [7] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *5th Int. Joint Conf. Artificial Intelligence*, pages 659–663, Cambridge, MA, 1977.
- [8] A. Basu and S. Licardie. Alternative models for fisheye lenses. *Elsevier Pattern Recognition Letters*, 16(4):433–441, 1995.
- [9] R. Bernstein. Digital image processing of earth observation sensor data. *IBM J. Res. Develop.*, 20:40–67, 1976.

- [10] Gunilla Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE trans. on PAMI*, 10:849–865, 1988.
- [11] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.
- [12] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on PAMI*, 26(9):1124–1137, 2004.
- [13] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.
- [14] John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [15] R. Carroll, M. Agrawala, and A. Agrawala. Optimizing content-preserving projections for wide-angle images. *ACM Trans. Graph*, 28(3), 2009.
- [16] P. Dagar and A. Danke. Property tax assessment system using GIS. In *Urban Planning*. http://www.gisdevelopment.net/application/urban/overview/urban_p005.htm, 2010.
- [17] P. A. Van den Elsen, J. B. A. Maintz, E.J.D. Pol, and M. A. Viergever. Automatic registration of ct and mr brain images using correlation of geometrical features. *IEEE Transactions on medical images*, 14(2):384–398, 1995.
- [18] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
- [19] R. O. Duda and P. E. Hart. Use of the hough transform to detect lines and curves in pictures. *Comm. ACM*, 15(1):11–15, 1972.
- [20] P. A. Van Den Elsen, E-J D. Pol, and M. A. Viergever. Medical image matching—a review with classification. *IEEE Eng. Med. Biol.*, pages 26–39, 1993.

- [21] J. Gluckman. Gradient field distributions for the registration of images. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 691–694, 2003.
- [22] R. C. Gonzalez and R. E. Woods. *Digital Image processing*. Prentice Hall International, second edition edition, 2002.
- [23] A. Ardeshir Goshtasby. *2-D and 3-D image registration for medical, remote sensing and industrial applications*. Wiley Press, 2005.
- [24] R. V. L. Hartley. Transmission of information. *Bell Systems Technical Journal*, 7.
- [25] C. Hughes, M. Glavin, E. Jones, and P. Denny. Review of geometric distortion compensation in fisheye cameras. In *IET ISSC'08*, pages 162–167, 2008.
- [26] C. Ishii, Y. Sudo, and H. Hashimoto. An image conversion algorithm from fish eye image to perspective image for human eyes. In *Proc. of the IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pages 1009–1014, 2003.
- [27] L. Junck, J. G. Moen, G. D. Hutchins, M. B. Brown, and D. E. Kuhl. Correlation methods for the centering, rotation, and alignment of functional brain images. *Journal of nuclear medicine*, 31:1220–1276, 1990.
- [28] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *Proc. ACM Transactions on Graphics, SIGGRAPH 2003*, 2003.
- [29] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.
- [30] J. le Moigne. Parallel registration of multi-sensor remotely sensed imagery using wavelet coefficients. In *Proceedings of the SPIE: Wavelet Applications*, pages 432–443, 1994.

- [31] S. Lhomme, Dong-Chen He, C. Weber, and D. Morin. A new approach to building identification from very-high-spatial-resolution images. *International Journal of Remote Sensing*, 30(5):1341–1354, 2009.
- [32] J. Liu, B.C. Vemuri, and J.L. Marroquin. Local frequency representations for robust multimodal image registration. *IEEE Transactions on Medical Imaging*, 21:462–469, 2002.
- [33] M.Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *CVPR'10*, pages 1696–1703, 2010.
- [34] D.G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, 1999.
- [35] J. Maantay and J. Ziegler. *GIS for the urban environment*. ESRI Press, 2006.
- [36] J. Mallon and P.F. Whelan. Precise radial un-distortion of images. In *Proc. of the IEEE Int. Conf. on Pattern Recognition*, pages 18–21, 2004.
- [37] T. Nathan Mundhenk, Michael J. Rivett, Xiaoqun Liao, and Ernest L. Hall. Techniques for fisheye lens calibration using a minimal number of measurements. In *Proc. SPIE Intell. Robot. Comput. Vis. Conf. XIX*, Boston, MA, 2000.
- [38] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transaction on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.
- [39] Josien P. W. Pluim, J. B. Antoine Maintz, and Max A. Viergever. Mutual information based registration of medical images: A survey. *IEEE Trans. on Medical Imaging*, 22(8):986–1004, 2003.
- [40] M. R. Sabuncu. *Entropy-based image registration*. PhD thesis, Dept. of Electrical Engineering, Princeton University, 2006.

- [41] R. Shams, P. Sadeghi, and R. A. Kennedy. Gradient intensity: A new mutual information based registration method. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR) Workshop on Image Registration and Fusion*, Minneapolis, MN., 2007.
- [42] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.
- [43] R. Turcajova and J. Kautsky. A hierarchical multiresolution technique for image registration. In *Proceedings of SPIE Mathematical Imaging: Wavelet Applications in Signal and Image Processing*, Colorado, 1996.
- [44] P. A. van den Elsen, E. J. D. Pol, T. S. Sumanaweera, P.F. Hemler, S. Napel, and J. Adler. Grey value correlation techniques used for automatic matching of ct and mr brain and spine images. *Visualization in Biomedical Computing*, 2359:227–237, 1994.
- [45] P. Viola. *Alignment by maximization of mutual information*. PhD thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1995.
- [46] Z. Wang, F. Wu, and Z. Hu. MSLD: A robust descriptor for line matching. *Pattern Recognition*, 42:941–953, 2009.
- [47] G. Wolberg and T. Boult. Separable image wrapping with spatial lookup tables. In *ACM SIGGRAPH '89*, pages 369–377, 1989.
- [48] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.
- [49] D. Zorin and A. H. Barr. Correction of geometric perceptual distortion in pictures. In *Proc. SIGGRAPH*, pages 257–264, 1995.
- [50] M. Zouqi, J. Samarabandu, and Y. Zhou. Automatic position registration of street-level fisheye images into aerial image using line structures and mutual information. In *Seventh Canadian Conference on Computer and Robot Vision*, Ottawa, ON, CA, 2010. IEEE.

- [51] M. Zouqi, J. Samarabandu, and Y. Zhou. Multi-modal image registration using line features and mutual information. In *International Conference on Image Processing*, Hong Kong, 2010. IEEE.

Vita

NAME: Mehrnaz Zouqi

PLACE OF BIRTH: Iran

YEAR OF BIRTH: 1977

POST-SECONDARY
EDUCATION
AND DEGREES: University of Western Ontario
London, Ontario
M.E.Sc, 2008

Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
B.E.Sc., 2001

HONOURS AND
AWARDS: Ontario Graduate Scholarship (OGS), 2012-2013.

NSERC Industrial Postgraduate Scholarship (IPS),
2009-2011.

MITACS Scholarship, 2009-2011.

Best Presentation Award, Western Engineering
Graduate Symposium, 2010.

Western Graduate Thesis Research award, 2008.

Precarn Scholarship, 2008.

RELATED WORK
EXPERIENCE:

Research Assistant, Dr. J. Samarabandu's Lab, ECE Dept.,
The University of Western Ontario, 2007-2013.

Graduate Teaching Assistant, ECE Dept.,
The University of Western Ontario, 2007-2012.

Control Engineer, Pars Arc Co., Iran, 2002-2005.

Software Developer, Iran Telecom. Research Center (ITRC),
2000-2002.

Research Assistant, Arvin Sakhtafzar, Iran, 1997-2000.

Publications:

- M. Zouqi, J. Samarabandu, Y. Zhou, "Fusion of GPS and Image Data for Accurate Geocoding of Street-level Fisheye Images," in 6th International Conference on Information and Automation for Sustainability (ICIAfS2012), Sep. 2012.
- M. Zouqi, J. Samarabandu, Y. Zhou, "Multi-modal image registration using line features and mutual information," in IEEE International Conference on Image Processing (ICIP2010), pp. 129-132, Sep. 2010.
- M. Zouqi, J. Samarabandu, and Y. Zhou, "Automatic position registration of street-level fisheye images into aerial image using line structures and mutual information" in Canadian Conference on Computer and Robot Vision (CRV2010), pp. 115-121, May 2010.
- R. Rodrigo, M. Zouqi, Z. Chen, and J. Samarabandu, Robust and efficient feature tracking for indoor navigation, IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 39, no. 3, pp. 658-671, June 2009.
- M. Zouqi, and J. Samarabandu, "2D Ultrasound Image Segmentation Using Graph Cuts and Local Image Features," in IEEE Symposium on Computational Intelligence for Image Processing (CIIP2009), pp. 33-40, April 2009.

- M. Zouqi, and J. Samarabandu, Prostate segmentation from 2-D ultrasound images using graph cuts and domain knowledge, in Canadian Conf. on Computer and Robot Vision (CRV2008), pp. 359-362, May 2008.
- M. Azizian, and M. Zouqi, "Improving back propagation algorithm for training of multi-layer perceptron neural network for random signals" in 3rd Iranian Students' Electrical Engineering Conference Transactions, 1999.