

MSc thesis in Geomatics

An image-based method  
for the pairwise registration of  
mobile laser scanning point clouds

Antria Christodoulou

October 2018



ON COVER:

The upper Figure illustrates two point cloud tiles before registration. The two point cloud tiles represent the same scan, but have been captured at different times from different view points. The one point cloud is illustrated with red color and the other point cloud with gray color. Corresponding objects from the two point clouds do not match. This can be clearly seen by observing the vertical structures (light poles). The lower Figure illustrates the point cloud tiles of the upper Figure after registration.

AN IMAGE-BASED METHOD FOR THE PAIRWISE REGISTRATION OF  
MOBILE LASER SCANNING POINT CLOUDS

A thesis submitted to the Delft University of Technology in partial fulfillment  
of the requirements for the degree of

Master of Science in Geomatics for the Built Environment

by

Antria Christodoulou

October 2018

Antria Christodoulou: *An image-based method for the pairwise registration of mobile laser scanning point clouds* (2018)

© This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit

<http://creativecommons.org/licenses/by/4.0/>.

ISBN: 978-94-6186-984-5

The work in this thesis was made in the:



Department of GIS Technology, OTB Research Institute for  
the Built Environment  
Faculty of Architecture  
Delft University of Technology

Company: CycloMedia Technology B.V.

Supervisors: Prof.Dr. Ir. P.J.M. van Oosterom  
Dr. R.Y. Peters

Ext. Supervisors: Ir. P. Joosten  
MSc. B. van Someren

Co-reader: Dr. Jinhu Wang

# ABSTRACT

In this thesis, an image-based method is proposed for solving the relative translation errors of 3D point clouds collected by Mobile Laser Scanning (MLS) techniques. The process of matching 3D point clouds is known as registration. Due to environment-depending limitations of the positioning component of MLS systems, when recording a certain scene more than once, point clouds recorded at different times tend to have mainly positioning distortions. Additionally, due to yaw-angle errors of the recording platform it is possible that the point clouds have small distortions around their Z axis or the point clouds are scanned with fuzziness. This project deals only with the translation errors.

Various methods can be found in the literature to perform pair-wise registration of point clouds. Commonly, the challenge of aligning 3D features is tackled in 3D. Only a few techniques for registering point clouds in 2D have being explored. However, the approach presented in this thesis uses the attributes of the 3D points to generate and match 2D-projections, by employing a simple correlation technique instead of matching in 3D. As a result, the developed method depends more on the number of pixels in the 2D-projections and less on the number of points in the point clouds. This leads to a more cost-efficient method in contrast to 3D registration techniques. The method uses this benefit to provide redundant translation parameters for each point cloud pair. Particularly, several images are created from each point cloud tile. The constructed images illustrate the density of the points, the intensity, the gradient of intensity, the depth, the gradient of the depth and the normal vectors of the points. As a result, the translation parameters are retrieved from the matching of various image-techniques and that is how redundant solutions are provided. Next, with the utilization of image-based evaluation criteria the reliable translation parameters are detected and only those are used to compute the final solutions. The reliable estimations are taken into account for the estimation of the final solutions. Since redundant solutions are provided, the confidence levels of each final estimation can be computed. In addition, an indication of robustness showing how many estimations where included for the computation of the final solution is included. As a result, the developed approach is capable of providing information about the precision and reliability of each pairwise registration. In such a way it is known which of the results can be used in a following step, such as a global registration. Furthermore, a 2D Gaussian elliptical fitting is used to obtain sub-pixel accuracy registration results, as the accuracy of the estimations is restricted to the pixel size of the generated images.

It is proven that the developed image-based registration method has the capability to produce reliable matches when there is at least some overlap between two overlapping point clouds and corresponding objects between the point clouds are distinct in pairs of 2D projections. The technique developed for the computation of sub-pixel accuracy results seems to have potential, but further improvement is required.



## ACKNOWLEDGEMENTS

I am grateful to all the people who have supported me along the way of completing this thesis!

First of all I would like to thank Peter Joosten and Berry van Someren from the company CycloMedia who were my daily supervisors. Even though I was not at the company's office every day, I could easily reach both just through a skype message.

Particularly, I would like to thank Peter for sharing with me all his knowledge about mobile laser scanning, mobile positioning and general geo-knowledge. Among other, I learned the basics about the adjustment theory of surveying data, I learned about the professors Baarda W. and Teunissen P.J.G. who are behind this theory, but also that in USA different states use coordinate reference systems with different foot size's (international foot and survey foot). I will not forget that he was answering my messages even on weekends.

I would like to thank Berry for helping me with all the ICT questions I had and for imparting his knowledge about the image processing. I really appreciate this since I have a different background. Moreover, I would like to thank Berry about the extensive feedback on my documents, even though we did always agree! I learned to think twice before I write something and be sure that I explain every time 'why'! I will definitely use that in my future writings. I will not forget that he had enough patient to talk to me in Dutch during breaks to help me practise learning the language.

I would like to thank my professor Peter van Oosterom for his guidance and for the long and fruitful discussions we had! Peter was there to remind me continuously that discrediting the point clouds by creating imagery might have its advantages, but accuracy is lost. Therefore, the proposed method was extended and a relevant solution to tackle the discrete accuracy was developed! I would also like to thank my second supervisor Ravi Peters for his feedback on my writings and for being there for me by immediately answering my emails and providing me with his advice!

Many thanks also to Arjen Swart, Bas Boom and Ruud Verwaal from the company Cyclomedia for the knowledge they shared with me. I am also grateful to Edward Verbree for guiding me in order to propose a solution for distributing the relative positioning error between overlapping point clouds and also to Mathias Lemmens for advising me with respect to the development of a sub-pixel accuracy matching method.

I would also like to thank my friend Lydia for her constant support during this project. Lydia was there to listen to the problems I was facing and just by discussing with her was enough to help my self reconsider and improve things!

A very special gratitude goes to Wim Boschman and Eric Hagemans who did not directly contributed to this project, but they are those who helped me move from my country and start in the Netherlands as an intern at the dutch Cadastre. Without them I wouldn't be were I am today!

I would also like to thank my parents Nikos and Maria, my siblings and my friends who, although remotely, were continuously supporting me during the thesis project and my studies in general.

And finally, last but by no means least I would like to give my biggest thank you to my boyfriend Yannick who was daily next to me to listen, to support, to encourage me!



# CONTENTS

1	INTRODUCTION	1
1.1	Problem statement and motivation	2
1.1.1	Problem origin	2
1.1.2	Problem statement	3
1.1.3	Motivation	5
1.2	Objectives & research question	5
1.3	Use case	6
1.3.1	Laser scanner	6
1.3.2	Point cloud data	7
1.3.3	Thesis outline	8
2	THEORETICAL BACKGROUND	9
2.1	Local point cloud registration	9
2.1.1	Local coarse registration and local fine registration	10
2.1.2	Fixed & moving point clouds	10
2.2	Global point cloud registration	11
2.3	Quality of local point cloud registration	11
2.4	Point clouds processing	12
2.4.1	Rejection of outlying points	13
2.4.2	Normal vectors	16
2.4.3	The Principal Component analysis method	17
2.5	Image processing	17
2.5.1	Image gradient	18
2.5.2	Sobel Operator	19
2.5.3	Gaussian Smoothing	20
2.5.4	Image registration	20
2.5.5	Template matching	20
3	RELATED WORK	23
3.1	3D local point cloud registration	23
3.1.1	Point clouds local registration with ICP	23
3.1.2	Point clouds local Registration with ICP variants	24
3.1.3	Local registration quality	26
3.1.4	Outlier rejection to minimize influence on registration	27
3.1.5	Conclusions	28
3.2	2D local point cloud registration	30
3.2.1	Image-based local point cloud registration	30
3.2.2	From 3D point clouds to 2D imagery	31
3.2.3	Sub-pixel accuracy	34
3.2.4	Conclusions	35
4	IMAGE-BASED POINT CLOUD LOCAL REGISTRATION APPROACH	37
4.1	Motivation	37
4.2	Method overview	38
4.3	Pre-processing	39
4.3.1	Outlier rejection	39
4.3.2	Normal vectors computation	42
4.3.3	Pairing overlapping point clouds	43
4.4	Conversion from 3D to 2D	44
4.4.1	Reduction of dimensions	44
4.4.2	Density images	50

4.4.3	Intensity images	52
4.4.4	Gradient of intensity images	53
4.4.5	Depth images	55
4.4.6	Gradient of depth images	57
4.4.7	Images based on normal-vectors	57
4.4.8	Images based on the components of the normal-vectors	59
4.5	Pairwise image registration	62
4.6	Image coordinates to space coordinates	65
4.7	Optimal Solution & Quality Evaluation	66
4.8	Sub-pixel accuracy	75
5	IMPLEMENTATION & EXPERIMENTS	85
5.1	Data-sets Used & Tools	85
5.1.1	Data-sets	85
5.1.2	Software & Hardware	85
5.2	Implemented prototype	87
5.2.1	Input data	87
5.2.2	Normal vectors	87
5.2.3	Reduction of dimensions	87
5.2.4	Image registration	88
5.3	Experiments	89
6	RESULTS & ANALYSIS	93
6.1	Main method results	93
6.1.1	Case 1: Big overlap, changes in the scene, uncrowded scene	93
6.1.2	Case 2: Big overlap, large offset, crowded scene	95
6.1.3	Case 3: Small overlap, presence of a bridge and a road	95
6.1.4	Case 4: Big overlap, large offset, distinct structures	96
6.1.5	Case 5: Absence of overlap, presence of tunnel	97
6.2	Sub-pixel accuracy results	99
6.3	Image-type suitability for registration	100
7	CONCLUSIONS & FUTURE WORK	105
7.1	Research Questions	105
7.2	Future Work	110
7.2.1	Recommendations to improve the method	110
7.2.2	Recommendations to extend the project	112
A	REFLECTION	121

# LIST OF FIGURES

Figure 1.1	A mobile laser scanning platform. . . . .	1
Figure 1.2	Multipathing and blockage of the navigation signals. . . . .	2
Figure 1.3	Offset in the horizontal direction between point clouds. . . . .	3
Figure 1.4	Effects of the yaw angle error on the point clouds. . . . .	4
Figure 1.5	Example in top view of the yaw angle error effects. . . . .	4
Figure 1.6	The laser scanner Velodyne HDL-32. . . . .	7
Figure 1.7	Example of a tile including millions of points. . . . .	7
Figure 1.8	Example of a tile including a few thousands of points. . . . .	8
Figure 2.1	Translation transformation of a point. . . . .	9
Figure 2.2	Local coarse registration, local fine registration and global registration. . . . .	10
Figure 2.3	Fixed and moving scans. . . . .	11
Figure 2.4	Moving objects present at the one point cloud, but not in the other overlapping point cloud. . . . .	13
Figure 2.5	Noise in the point clouds when objects are occluded by other objects. . . . .	14
Figure 2.6	The concept of the Local Outlier Factor method for outlier rejection and its main components. . . . .	15
Figure 2.7	The Local Reachability Density of a point. . . . .	15
Figure 2.8	Normal vectors of 3D points. . . . .	17
Figure 2.9	Image gradient in 2D and 3D. . . . .	18
Figure 2.10	A simple example of the image gradient. . . . .	19
Figure 2.11	Image registration example. . . . .	20
Figure 2.12	The main concept of the template matching technique. . . . .	21
Figure 2.13	The size of the score map array is based on the possible overlays between the two images that must be registered. . . . .	21
Figure 2.14	An example output of a template matching technique. . . . .	22
Figure 3.1	The concept of the Iterative Closest Point algorithm. . . . .	24
Figure 3.2	Limitations of ICP. . . . .	25
Figure 3.3	A 2D attribute space for outlier rejection, and a comparison of a Classical and a robust covariance ellipse. . . . .	28
Figure 3.4	Difference between the density of inliers and outliers. . . . .	29
Figure 3.5	A feature-based point cloud registration method. . . . .	30
Figure 3.6	A depth image example. . . . .	31
Figure 3.7	An example of a perspective intensity image in grayscale. . . . .	32
Figure 3.8	Definition of the bearing angle. . . . .	32
Figure 3.9	Bearing angle images highlighting different directions. . . . .	33
Figure 3.10	An example of an xy-histogram. . . . .	33
Figure 3.11	An example of an z-histogram. . . . .	34
Figure 3.12	Second order polynomial to interpolate pixel values. . . . .	34
Figure 4.1	Main steps of the proposed method. . . . .	38
Figure 4.2	The local density of points as a parameter for outlier rejection. . . . .	39
Figure 4.3	Comparison of relative and absolute threshold. . . . .	41
Figure 4.4	The perspective and the orthogonal projection. . . . .	44
Figure 4.5	A perspective and an orthogonal projection. . . . .	44
Figure 4.6	The reduction of dimensions. . . . .	46
Figure 4.7	A point cloud tile viewed from the three planes used for the orthographic projections. . . . .	46
Figure 4.8	A benefit of having a triple of 2D projections. . . . .	47
Figure 4.9	The combined boundaries of a point cloud pair for the XZ-plane. . . . .	48

Figure 4.10	Binning 3D points of a point cloud in a 2D grid. . . . .	49
Figure 4.11	Binning 3D points of an overlapping point cloud in a 2D grid. . . . .	49
Figure 4.12	Example of images based on the density. . . . .	51
Figure 4.13	The drawback of the density images . . . . .	51
Figure 4.14	Example of images based on the intensity. . . . .	52
Figure 4.15	The reason why images with the gradient of intensity are required. . . . .	53
Figure 4.16	The gradient of intensity images do not solve the drawback of the intensity images. . . . .	54
Figure 4.17	Decision for the kernel size of the Sobel operator. . . . .	55
Figure 4.18	Example of images based on the gradient of intensity. . . . .	55
Figure 4.19	Example of images based on the depth. . . . .	56
Figure 4.20	Example of images based on the gradient of the depth. . . . .	57
Figure 4.21	Projections with the normal vectors. . . . .	59
Figure 4.22	Example of images based on the X component of the normal vectors. . . . .	60
Figure 4.23	Example of images based on the Y component of the normal vectors. . . . .	61
Figure 4.24	Example of images based on the Z component of the normal vectors. . . . .	61
Figure 4.25	The 27 produced image pairs from each overlapping point cloud pair. . . . .	62
Figure 4.26	Increasing the search space between the images to perform template matching. . . . .	64
Figure 4.27	Two score maps resulted from matching a density XY-projection pair and the final score map. . . . .	65
Figure 4.28	Detecting local maxima in a score map to judge its quality. . . . .	69
Figure 4.29	Quality evaluation of the results. . . . .	71
Figure 4.30	Why a sub-pixel accuracy method is required. . . . .	76
Figure 4.31	The area on a score map where a sub-pixel accuracy method may be applied. . . . .	77
Figure 4.32	Horizontal and vertical fitting on a score map. . . . .	77
Figure 4.33	Second order polynomial fitting that take into account 4 neighbouring pixels to the peak pixel. . . . .	78
Figure 4.34	Second order polynomial fitting that take into account 2 neighbouring pixels to the peak pixel. . . . .	79
Figure 4.35	According to the input parameters different orders of polynomials are fitted best to the score maps. . . . .	79
Figure 4.36	If the input parameters are constant, then polynomials of the same order fit best to different score maps. . . . .	79
Figure 4.37	High resolution score maps in 2D and in 3D. . . . .	80
Figure 4.38	An example of a 2D Gaussian and a 2D Laplacian Probability Density Function. . . . .	81
Figure 4.39	Equations of a bivariate Laplace and a 2D Gaussian function. . . . .	81
Figure 4.40	2D elliptical Gaussian fit on similarity scores around the peak. . . . .	83
Figure 5.1	Cartesian and computer coordinate system. . . . .	88
Figure 5.2	Execution time of algorithm's main steps when using different grid cell sizes. . . . .	90
Figure 5.3	Internal reliability and standard deviation of the X translation parameters as estimated for different grid cell sizes. . . . .	91
Figure 5.4	Internal reliability and standard deviation of the Y translation parameters as estimated for different grid cell sizes. . . . .	91
Figure 5.5	Internal reliability and standard deviation of the Z translation parameters as estimated for different grid cell sizes. . . . .	91
Figure 6.1	An example of relatively large overlap between point clouds. . . . .	93
Figure 6.2	An example point cloud pair before and after registration. . . . .	94

Figure 6.3	Quality measures of the point cloud pair illustrated in Figure 6.2. . . . .	94
Figure 6.4	An example point cloud pair before and after registration. . . . .	95
Figure 6.5	Quality measures of the point cloud pair illustrated in Figure 6.4. . . . .	95
Figure 6.6	An example point cloud pair before registration. . . . .	96
Figure 6.7	An example point cloud pair after registration and its quality measures. . . . .	96
Figure 6.8	An example point cloud pair before and after registration. . . . .	97
Figure 6.9	Quality measures of the point cloud pair illustrated in Figure 6.8. . . . .	97
Figure 6.10	An example point cloud pair before and after registration. . . . .	98
Figure 6.11	A panoramic and an aerial image to get more insight of the point clouds. . . . .	98
Figure 6.12	An example point cloud pair before and after registration. . . . .	98
Figure 6.13	Quality measures of the point cloud pair illustrated in Figure 6.10. . . . .	99
Figure 6.14	Sub-pixel accuracy method results. . . . .	100
Figure 6.15	Redundancy number of the estimated X translation errors per image type. . . . .	101
Figure 6.16	Redundancy number of the estimated Y translation errors per image type. . . . .	101
Figure 6.17	Redundancy number of the estimated Z translation errors per image type. . . . .	102
Figure 6.18	Illustrating the reason why the normal vectors do not have the best performance. . . . .	103
Figure 6.19	Score maps resulted from the matchings of XZ and YZ density projections. . . . .	103
Figure 7.1	Applying a (Gaussian) fitting to all the peaks formed by the similarity values in a score map. . . . .	109



## LIST OF TABLES

Table 4.1	A comparison between the perspective and orthographic projections concerning the image registration. . . . .	45
Table 4.2	Example of images based on the normal vector. . . . .	58
Table 4.3	The table shows the redundant solutions are decreased from 27 per translation parameter ( $X, Y, Z$ ) to 9 per translation parameter. . . . .	70
Table 4.4	The table is an extension of the example shown in Table 4.3. It illustrates the internal reliabilities of the estimations outputted from each image type. . . . .	72
Table 4.5	The table is an extension of the example shown in Table 4.3. It illustrates the overall internal reliability of each estimation. . . . .	73
Table 4.6	The table shows the (external) reliability weights given to image registration results according to the image area ( $m^2$ ) that adopts information from the 3D points. . . . .	74
Table 4.7	The table is an extension of the example given in the table 4.3. It shows how the precision is computed for each final translation parameter. The final translation parameters are symbolized as $Ov.\bar{X}$ , $Ov.\bar{Y}$ , $Ov.\bar{Z}$ . The precision of the final parameters are based on the computation of the standard deviation. The standard deviations are symbolized as $Ov.\sigma_X$ , $Ov.\sigma_Y$ and $Ov.\sigma_Z$ , and they are computed from the accepted solutions retrieved per image type. If an image type does not provide any solutions, then only the remaining solutions are considered. . . . .	75
Table 5.1	The table illustrates information about the data-sets used. The first column is a count for the scenes of overlapping point cloud tiles. Each scene refers to a square area of $250m^2$ . The number of existing overlapping point cloud tiles for each scene is shown in the second column. The number of the constructed point cloud pairs for each scene is shown in the third column. For the calculation of the number of point cloud pairs in each scene see Section 4.3.3. The number of the image pairs constructed for each scene is shown in the fourth column. This is the result of the number of point cloud pairs per scene multiplied by 27, which is the number of image types created per point cloud. . . . .	86



# LIST OF ALGORITHMS

4.1	LOF - Relative threshold . . . . .	40
4.2	LOF - Absolute Threshold . . . . .	40
4.3	Normal vectors on 3D points . . . . .	42
4.4	Binning the 3D points in a 2D grid . . . . .	49



# ACRONYMS

UAV	Unmanned Aerial Vehicle	1
MLS	Mobile Laser Scanning	1
LiDAR	Light Detection And Ranging	1
GNSS	Global Navigation Satellite System	1
IMU	Inertial Measurement Unit	1
DR	Dead-Reckoning	2
ICP	Iterative Closest Point	5
FOV	Field Of View	6
LOF	Local Outlier Factor	14
PGO	Pose Graph Optimization	112
MSE	Mean Square Error	23
ICCP	Iterative Closest Compatible Point	25
RICP	Robust Iterative Closest Point	25
LMedS	Least Median Squares	25
LTS	Least Trimmed Squares	25
ICPIF	Iterative Closest Point using Invariant Features	26
k-NN	<i>k</i> -Nearest Neighbors	14
lrd	Local Reachability Density	14
COF	Connectivity Outlier Factor	27
MCD	Minimum Covariance Determinant	27
PCA	Principal Component Analysis	17



# GLOSSARY

## B | G | I | K | L | O | P | T

### B

**Breakdown Point** A breakdown point defines the degree of robustness of an estimate in the presence of outliers. It measures the amount of outliers an estimate can handle before giving a spoiled estimate [[Hampel, 1971](#)]. 26

### G

**geodetic observations** Observations such as coordinates, distances and angles between points measured to derive the coordinates of other points of interest. The data are measured with surveying techniques by using instruments such as total stations for angles and distances, optical levelling instruments for elevations, and Global Navigation Satellite System (GNSS) receivers to collect coordinates of (permanent) reference stations or other points.. 11, 12, 66, 67

**global registration** Global, multiview, or absolute registration refers to the integration of all the retrieved point clouds in a common reference system. To do so, all the the individual relative registrations need to be fused into a single point cloud [[Sanchez et al., 2017](#)]. 5, 109, 112

### I

**image pair** An image pair is a pair of images which originates from a [point-cloud pair](#). The two images are the outputs of projecting the two corresponding point clouds with the same method. The values of the pixels of the two images contain the same information such as intensity, depth etc.. 30, 47, 50, 51, 62

### K

**k-distance** The  $k$ -distance of an observed point is the distance within which at least the  $k$  points can be found.. 14, 15

**kd-tree** A  $k$ -dimensional tree is a data structure that organizes a dataset as a tree, which was developed by [Bentley \[1975\]](#). The  $k$  refers to the dimensionality of the space. Thus for point clouds the dimensionality of the kd-tree is three. To store the data, a random attribute among the  $X, Y, Z$  is peaked, and then for that attribute the median is found. The median is used to split the dataset, thus the points are split in the middle of the selected attribute. The same process repeats recursively until the algorithm ends up with a predetermined number of points left in each branch of the tree. Then this structure can be used to find the  $k$ -nearest neighbors. The leaf of the tree that an examined point is stored is found and then the neighbours are searched within that leaf. It is possible that a point which is a nearest neighbor it is not detected as a neighbor because it is stored in another branch [[Lavrenko, 2015](#)]. Nevertheless, it is a data structure that makes the search for nearest neighbors fast, as the neighbors are searched in a small search space.. 86

### L

**LAS format** LAS is a format for storing data retrieved from Light Detection And Ranging processes. LAS files are binary [Brown and Butler, 2012] and open, which means that there are no restrictions regarding their use and users [Isenburg, 2015]. LAS files are decompressed LAZ format files [Isenburg, 2015].. xxii, 85

**LAZ format** It is a compressed LAS format and it is an open format [Isenburg, 2015].. xxii, 7, 85

**local registration** Local, pairwise, or relative registration refers to the estimation of the relative transformation required to match point-cloud pairs [Magnusson et al., 2007].. 5

## O

**overlapping point clouds** This is similar to the term point-cloud pair with the difference that overlapping point clouds may refer to more than two point cloud sets.. 3, 5, 24

## P

**point-cloud pair** A point cloud pair refers to two point cloud tiles (point-cloud sets) that overlap as they cover the same scene, but captured at different times from different observation points. The overlap between the overlapping point clouds may vary.. xxi, xxii, 11, 23, 25, 26, 28, 37, 38, 48, 50, 67

**point-cloud set** In the case of this project a set of 3D points refer to the points included in a tile of 50 by 50 meters.. xxii, 39

## T

**time-of-flight** This is a technology with which the distance from a laser sensor to the surrounding objects is measured. Specifically, the time that a pulse takes to travel from a laser sensor to an object and back is measured. The distance is then computed as a multiplication of the travelling time with the speed of light divided by two as the distance is measured once towards the object and once from the object to the laser detector [Lemmens, 2011b].. 6

# 1

## INTRODUCTION

Point cloud data is an important source of 3D spatial information, which is growing in popularity and size [Van Oosterom et al., 2014]. Point cloud acquisition techniques, such as laser scanning techniques employ laser scanners mounted on static, mobile, or airborne platforms. These techniques are often preferable over traditional land surveying methods. The reason is that vast amounts of 3D data can be acquired in a minimal amount of time and with less manpower [Mendenhall, 2014]. Moreover, the 3D points have a high density that allows for a wide variety of applications. For instance, point cloud data can be used for deformation monitoring. This can be achieved by placing targets on the points of interest, such as bridges' foundations and railway lines, collecting point clouds periodically and measuring the displacement of the target points. Point clouds can be used also for real estate appraisal by assessing the condition, the neighbourhood and the location of a property. Another example is when point clouds are collected by Unmanned Aerial Vehicle (UAV)s for disasters management. In such cases algorithms are developed for the classification of the buildings' damage.

Laser scanners mounted on aerial platforms are particularly useful for recording terrain, forests and the outline and roof shape of buildings. Laser scanners mounted on static platforms are useful for objects that can be viewed from the ground and for small scale projects. These can be for example the scanning of the interiors and the facades of some buildings. In cases of large scale projects where there is a demand of immense street-view information acquisition, Mobile Laser Scanning (MLS) techniques are employed [Haala et al., 2008]. MLS is a technology that incorporates Light Detection And Ranging (LiDAR) sensors, Global Navigation Satellite System (GNSS) receivers and Inertial Measurement Unit (IMU) in a system mounted on a mobile platform (Figure 1.1). By using MLS the collection of 3D point data can be achieved rapidly and cost effectively [Kaartinen et al., 2012].

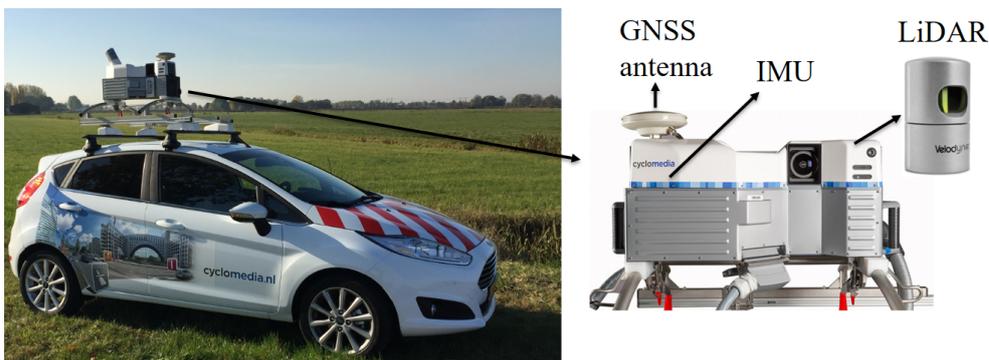


Figure 1.1: A Light Detection And Ranging (LiDAR) sensor, a Global Navigation Satellite System (GNSS) receiver and an Inertial Measurement Unit (IMU) deployed on a mobile platform for the purpose of street view geo-referenced laser scanning. The Figure depicts a MLS system as configured and mounted to a vehicle of the company CycloMedia Technology B.V.

## 1.1 PROBLEM STATEMENT AND MOTIVATION

### 1.1.1 Problem origin

The focus of this project is on point clouds retrieved from [MLS](#) techniques. The integration of observations from [GNSS](#) and [IMU](#) mounted on recording vehicles determine the vehicle's position and orientation [[Haala et al., 2008](#)]. By knowing the position and orientation of the mobile platform the coordinates of the scanned points can be derived as a function of distance. Namely, the [LiDAR](#) sensor mounted on the mobile platform records distances between itself and the scanned 3D environment.

The [GNSS](#) receiver on the platform can provide continuous positioning of good quality as soon as it receives direct signals (Figure 1.2) from four navigation satellites. Four range measurements between the receiver and the satellites are needed for the determination of the four unknowns. The first three unknowns are the X, Y, Z of the receiver's location. The fourth one is the time offset between the clock of the receiver and the clocks of the satellites [[Lemmens, 2011a](#)].

When the laser scanners provide data with accuracy of centimeter order, the accuracy of the retrieved point clouds from a [MLS](#) technique depends only on the accuracy of the integrated navigation solution [[Puente et al., 2011](#)]. Nevertheless, the [GNSS](#) reception may be limited due to dense blocks of high structures that surround the streets (Figure 1.2). Particularly, the [GNSS](#) signals may be reflected by neighbouring structures leading to longer signal traveling times (Figure 1.2) and potentially causing large errors in the positioning [[Shetty, 2017](#)]. In other cases, the signals of the satellites may be completely blocked. Thus the number of the available satellites may not be sufficient for the determination of the position.

As a result of the possible lack of [GNSS](#) reception, the navigation solution of the [MLS](#) system depends on the [IMU](#). An [IMU](#) computes position while the platform moves by using accelerometers. It also uses gyroscopes to measure the angular orientation of the scanner relative to the ground. The gyroscopes are additionally useful to discard effects of the gravity vector in the measured linear acceleration of the vehicle [[Levi and Judd, 1996](#)]. An [IMU](#) operates on the basis of the method Dead-Reckoning ([DR](#)), where every position is based on displacements from an initial known position [[Levi and Judd, 1996](#)]. Consequently, potential positioning errors are accumulated, leading to degraded accuracy of the provided positioning [[Barshan and Durrant-Whyte, 1995](#)].

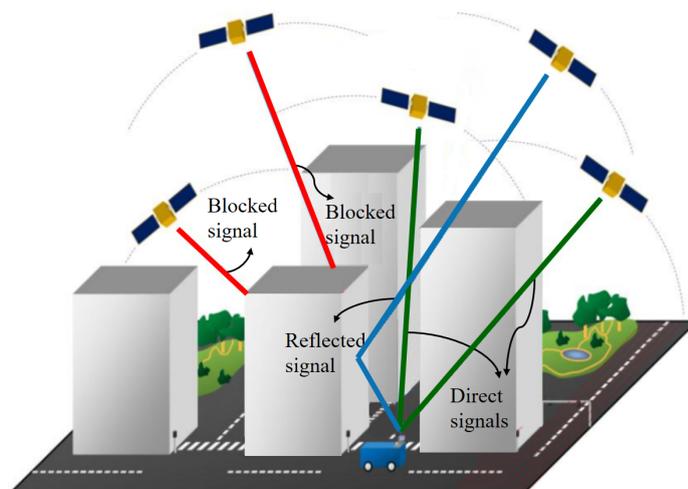
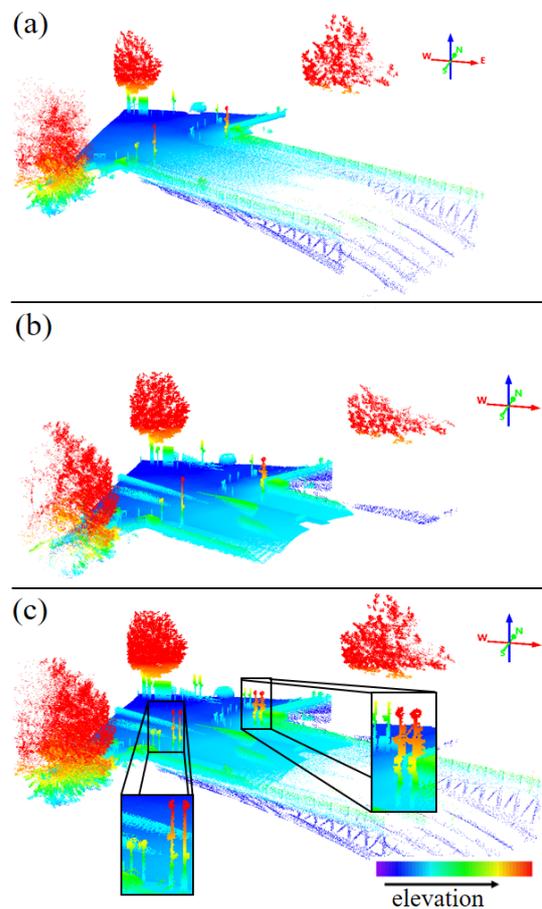


Figure 1.2: In environments with high-rise buildings there may be limited visibility of the signals from the navigation satellites. The signals may be blocked or reflected on neighbouring structures. (Modified figure from [[Kukko et al., 2012](#)]).

Apart from the positioning errors due to the limited GNSS visibility in specific environments, it is also possible to have orientation errors around the Z-axis of the mobile laser scanning platform. Namely, errors of the yaw angle. This is more likely to happen when the driver of the recording vehicle takes successively the same direction in turns. Since the orientation is determined by the IMU with which errors are propagated, orientation errors around  $\frac{1}{10}^{th}$  of a degree may be noticed [Joosten, 2017].

### 1.1.2 Problem statement

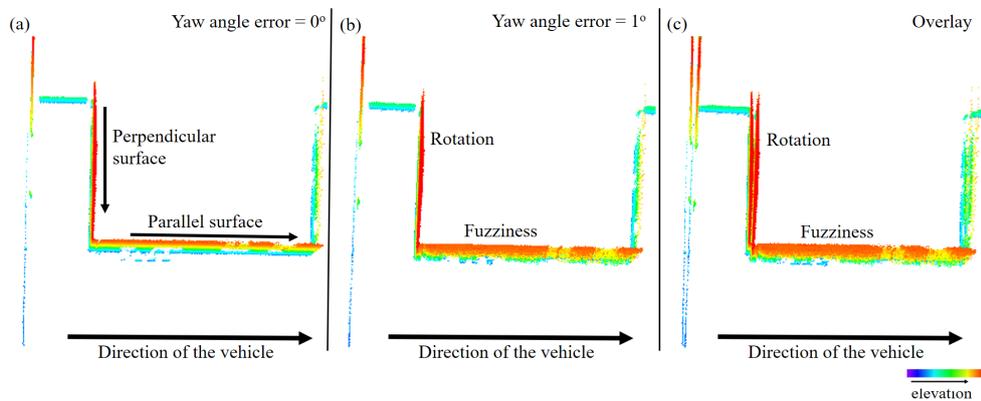
The lack of GNSS reception when scanning in environments that restrict the GNSS visibility and the probable error in the yaw angle of the recording platform can be easily noticed at the recordings of street-junctions. In such cases, the vehicle needs to return to a previously scanned region so as to re-scan and retrieve a complete scenery of the surroundings. Thus, overlapping point clouds are obtained. If the quality of the positioning is poor then corresponding points of the overlapping point clouds may have offsets in the X, Y, Z coordinates. Figure 1.3 illustrates two overlapping point clouds where the offset in the horizontal direction is conspicuous.



**Figure 1.3:** (a) Square point cloud tile of  $250m^2$ . (b) A second point cloud tile that captures the same region as (a) recorded at a different time. (c) The point clouds (a) and (b) do not match when are superimposed due to positioning errors. The offsets in the horizontal direction are especially visible at the magnified parts depicting electricity and sign poles.

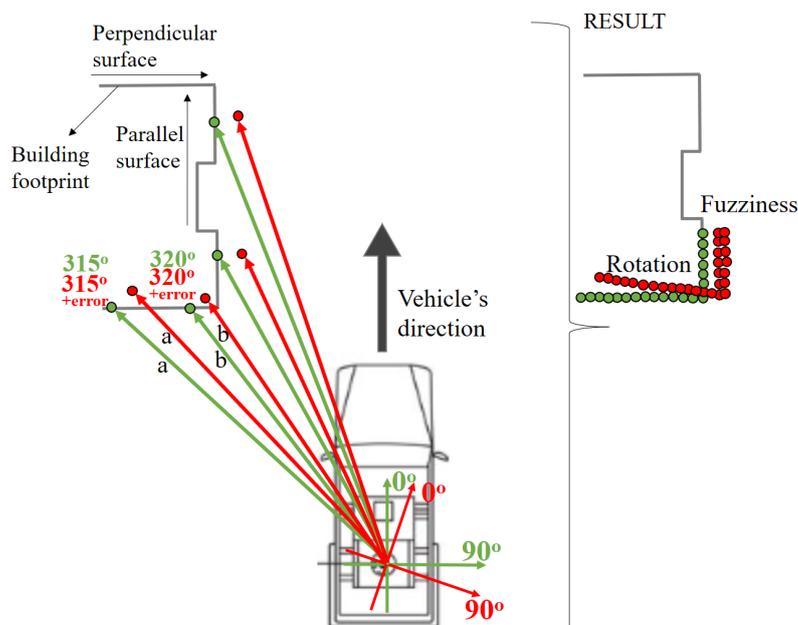
If there are yaw angle errors when recording a specific scene, then the collected points may be tilted or otherwise distorted. The effects of the rotation errors depend on the orientation of the 3D objects and the direction of the vehicle. Particularly, if the vehicle moves perpendicular to planar surfaces, then the surfaces are

represented tilted. If the vehicle moves parallel to planar surfaces, then the surfaces in the point clouds are represented fuzzed. Since the rotation errors are small (around  $\frac{1}{10}^{th}$  of a degree), it is not easy to visualize the problem. Therefore a simulation is provided with  $1^\circ$  artificial yaw angle error of the laser scanning platform as illustrated in Figure 1.4.



**Figure 1.4:** a) Top view of a point cloud scanned with no yaw angle error of the scanning platform. b) Top view of a point cloud scanned with artificial  $1^\circ$  yaw angle error of the scanning platform. It can be seen that the surfaces that are parallel to the trajectory of the recording vehicle are characterized by fuzziness. c) Overlay of (a) and (b). It can be seen that the surfaces that are perpendicular to the trajectory of the vehicle are tilted.

Figure 1.5 is given in order to understand the effects of the yaw angle errors. Everything that is coloured with green indicates that the recording system does not have a yaw angle error. Everything that is coloured with red indicates that there is a yaw angle error. A vehicle scans the surroundings. Firstly, a surface which is perpendicular to the direction of the vehicle is detected. A point which would be scanned at  $315^\circ$ , is scanned at  $315^\circ$  plus the degrees of the error. A point which would be scanned at  $320^\circ$ , is scanned at  $320^\circ$  plus the degrees of the error.



**Figure 1.5:** Example of the yaw angle error effects from a top view. Everything that is coloured with green indicates that there is no yaw angle error. Everything that is coloured with red indicates that there is a yaw angle error.

The laser beam coming from the platform with wrong orientation around the Z axis has exactly the same distance as the laser beam coming from the platform with no orientation errors. For example, the laser beam at the  $315^\circ$  is marked with an ' $a$ ' which indicates the laser beam's distance from the platform to the hit surface. The laser beam at the  $315+^\circ$  it is also ' $a$ ' meters long. Consequently, the point that is scanned at  $315+^\circ$ , apart from the fact that it is tilted, it is also captured slightly at the right of the point which is scanned with no error. The surface which is parallel to the direction of the vehicle appears fuzzy when there are orientation errors at the moving platform. That is because the laser beams with orientation error have exactly the same distance as those without error, as explained. At the right part of the figure the result is illustrated, where the dots represent the scanned points from the two cases.

Scaling errors are not encountered often, but in theory, when performing [MLS](#) for a long time without strong GNSS reception, the scale factor of the point clouds will not be equal to 1. For this project, it is assumed that possible scaling errors are negligible.

### 1.1.3 Motivation

The integration of all the retrieved point clouds in a common reference system is required after the recording of an area with a [MLS](#) system. This process is known as [global registration](#). Prior to that it is necessary to perform relative alignment of all the overlapping point clouds so that possible errors (as explained in Section 1.1.2) are minimized. This process is known as [local registration](#). These two steps constitute the 3D registration approach.

Locally and ultimately globally registered point clouds result in highly detailed and measurable 3D models that describe the geometry and topology of the objects in the real world. These models can then be used for several purposes such as object recognition, cultural heritage modeling [[Sanchez et al., 2017](#)], vehicle and pedestrian navigation, location based services [[Kaartinen et al., 2012](#)], disaster management, spatial analysis or even as alternatives to surveying processes such as coordinate extraction of cadastral parcels.

## 1.2 OBJECTIVES & RESEARCH QUESTION

The focus of this project is to solve the translation errors of overlapping point cloud pairs. Pairs of point clouds consist of two tiles of points that (partially) overlap because they represent the same scene. The point cloud tiles that overlap recorded from different observation points at different epochs. Many researchers employ a variant of the most commonly used algorithm for the local registration of [overlapping point clouds](#); the Iterative Closest Point ([ICP](#)) algorithm. These modified [ICP](#) methods deal with the limitations of [ICP](#) to compute pairwise registrations of higher quality. Despite some improvements to the main method, registration approaches that use the concept of [ICP](#) are sensitive to point cloud pairs in which not all the points in the one scan have a correspondences in the other scan [[Huang et al., 2017](#)]. They also have problems with point cloud pairs of which the initial positions of the two point clouds are not close to the required matching position [[Shetty, 2017](#)]. Moreover, [ICP](#)-based methods require computationally expensive and extensive search of point correspondences between the point clouds [Godin et al. \[1994\]](#). A thorough explanation is found in Section 3.1.

Given the limitations of [ICP](#) and given that the execution time of 3D registration methods is highly dependent on the number of points, the problem is approached differently in this research. The proposed approach uses the attributes of the 3D points to generate and match 2D-projections, in order to retrieve the transformation parameters that register pairs of 3D point clouds. In such a way the method will be

more dependant on the number of pixels in the 2D-projections and less on the number of points in the point clouds. This implies a more computationally efficient method in comparison with the 3D registration techniques. Therefore, the main research question for this thesis is formulated as follows:

*To which extent is it possible to automatically, reliably, precisely and efficiently align mobile laser scanning data relatively, using an image-based technique?*

To answer this question the following sub-questions are relevant:

1. How to deal with the outlying points in the data so that they do not affect the registration?
2. How to generate images from the point clouds and what kind of images so that the point clouds are best described?
3. How to compute the translation parameters that align relatively 3D overlapping point cloud pairs by matching 2D imagery?
4. How to assess reliability, precision and efficiency of the developed method automatically?
5. The accuracy of the transformation estimations will be restricted to the grid cell size of the generated imagery. How to retrieve a sub-pixel accuracy results to improve the drawback of discrete grid cell size in the imagery?

To summarize, this thesis focuses on how to match pairs of overlapping point clouds using images as an intermediate step. Specifically, the aim is to retrieve the translation parameters to minimize the translation errors which are mostly present. The effects on the point clouds from potential orientation errors in the yaw angle of the mobile platform are not solved. Furthermore, this project does not deal with the development of a global registration solution in order to align all the scans from an area in one common reference system.

## 1.3 USE CASE

### 1.3.1 Laser scanner

This project held in cooperation with the company CycloMedia Technology B.V. and the point cloud data-sets used are provided by the company. The employed laser scanner for the collection of the 3D points is a Velodyne HDL-32E. The head of the scanner rotates continuously about the system's central vertical axis to deliver a 360° horizontal Field Of View (FOV) [Chan and Lichti, 2015]. The rotation speed capability is from 5Hz to 20 Hz [Velodyne LiDAR, Inc., 2010]. It utilizes 32 laser-detector pairs aligned from +10° to -30° [Velodyne LiDAR, Inc., 2010] providing a vertical FOV of approximately 40 ° (Figure 1.6). The scanner has the potential to generate about 700.000 points per second [Velodyne LiDAR, Inc., 2012], capability of measurement range from 5 centimeters to 100 meters [Velodyne LiDAR, Inc., 2010] of which the usable range is up to 70m [Velodyne LiDAR, Inc., 2012] and a typical accuracy of +/-2cm. In addition, Velodyne HDL-32E operates with the *time-of-flight* principle [Chan and Lichti, 2015].

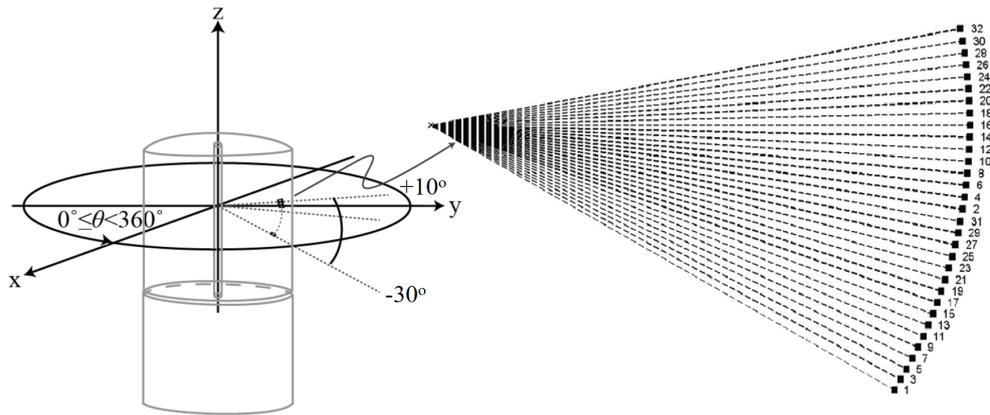


Figure 1.6: The figure illustrates how the laser beams are configured in the Velodyne HDL-32 laser scanner. Thirty-two radially oriented lasers are embedded inside the scanner from  $+10^\circ$  to  $-30^\circ$ . The numbering on the right of the lasers refers to their labeling given by the manufacturer. (Modified figure from [Chan and Lichti, 2015]).

### 1.3.2 Point cloud data

Point Cloud data is continuously collected during the mobile scanning process. The recorded 3D points are stored in tiles of 50m by 50m in the [LAZ format](#). For every tile of points there exists a corresponding LAZ file which contains the observation point of each point in the tile. In other words, the trajectory of the vehicle is provided as a 1 to 1 relationship with the recorded points.

Based on experience (CycloMedia, 2018), but also based on observations of the data, the translation errors between overlapping point cloud tiles range from millimeters to meters. Millimeter errors cannot be detected since the accuracy of the laser scanner is 2cm. Translation errors that reach meters are encountered when the GNSS reception from the position of the recording vehicle is significantly poor.

When the vehicle is fairly close to the scene a tile may contain millions of points, as illustrated in the Figure 1.7. In contrast, a tile may contain only a few thousands of points or even less in cases where the vehicle was recording a scene from far away (Figure 1.8).

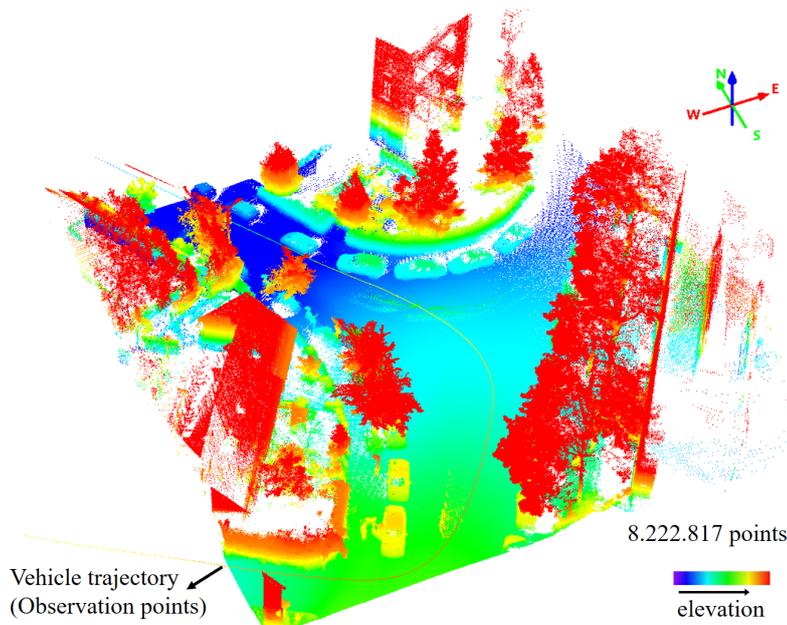
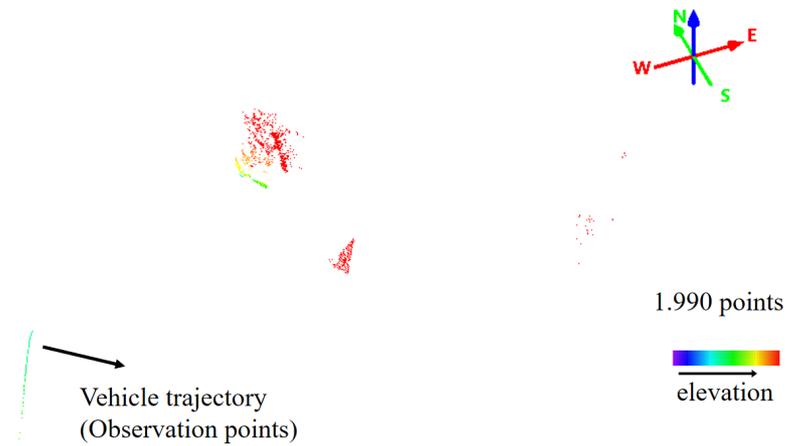


Figure 1.7: Example of a square point cloud tile of  $250m^2$  including millions of points.



**Figure 1.8:** Example of a square point cloud tile of  $250m^2$  including less than two thousands of points. The point cloud set illustrates the same scene as the tile in Figure 1.7, but recorded from different observation points.

### 1.3.3 Thesis outline

The rest of the thesis is structured as follows:

- A list of acronyms and a glossary explaining terms used in the thesis are given in pages [xix](#) and [xxii](#), correspondingly. Acronyms and words written in blue in the documents can be found in the aforementioned sections.
- The chapter [2](#) introduces the terminology used throughout this thesis and describes techniques and methods used for the development of the proposed algorithm.
- The chapter [3](#) describes and analyses approaches used in the bibliography for the relative registration of 3D point clouds. Mainly the point cloud registration is solved in 3D, but also some 2D techniques are presented. The advantages and disadvantages of those approaches are discussed and some relevant conclusions with respect to their performance are drawn.
- The chapter [4](#) motivates the development of an image-based point cloud registration method. A thorough description of the developed method follows, while every step taken is motivated.
- The chapter [5](#) provides information about the implementation details of the developed algorithm. It also contains experiments performed to judge the effectiveness of the proposed approach.
- The chapter [6](#) provides visual and arithmetical results of a number of point cloud registrations and analyzes their quality. It also includes results from the proposed sub-pixel accuracy method.
- The chapter [7](#) contains the answers to the research questions, the conclusions drawn regarding the performance of the developed method and the contributions of this work. It also contains recommendations on the aspects that could be improved and suggestions on how the specific project could be extended.
- Lastly, in the appendix [A](#) a reflection on the value of the graduation project in the larger social and scientific framework is given.

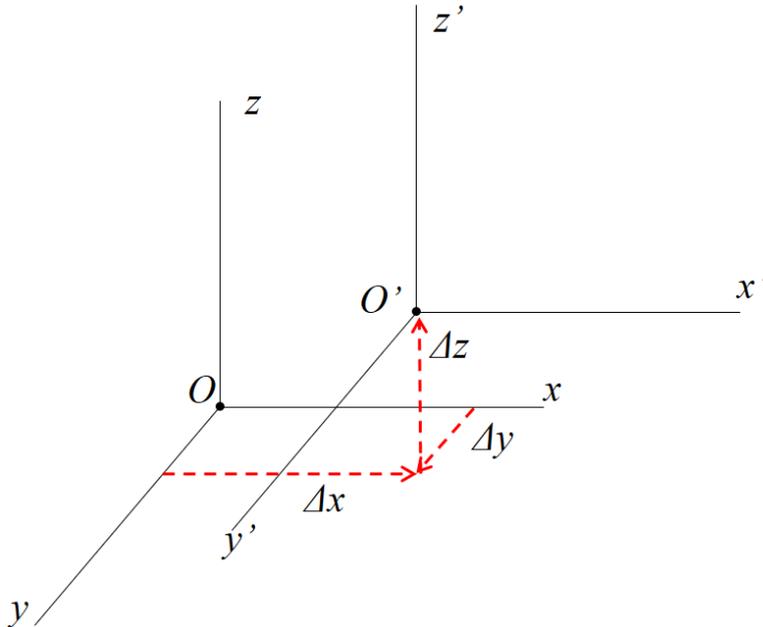
# 2

## THEORETICAL BACKGROUND

This chapter introduces the terminology used throughout this thesis and describes techniques and methods used in the proposed technique. Initially, the terms local and global registration are explained. Next, definitions relevant to the quality of a local point cloud registration method are given. Lastly, point cloud and image processing techniques are discussed.

### 2.1 LOCAL POINT CLOUD REGISTRATION

Local, relative or pairwise registration refers to the process of aligning pairs of point clouds or any other data which share common information. This is done by estimating the transformation parameters needed to match one with another [Magnusson et al., 2007]. In the case of mobile scanned point clouds the transformation parameters depend on the circumstances with which the point clouds are retrieved. As for this project the focus is on the translation errors resulted from poor positioning, finding the relative translation parameters is required (Figure 2.1). A translation constitutes a geometric transformation which moves each point of a point cloud tile by the same distance in a given direction. For example, if an X translation equals to 20cm, a Y translation equals to 60cm and a Z translation equals to 1.5m, then each point of one point cloud will be shifted 20cm in X, 60cm in Y and 1.5m in Z. Then the shifted point cloud could be matched to the reference point cloud.



**Figure 2.1:** The example illustrates a relative transformation where only translation is required to match two corresponding points from different data-sets. The displacements  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$  are applied onto the point  $O$  so as to match it with point  $O'$ .

### 2.1.1 Local coarse registration and local fine registration

Registering point clouds locally is in some cases split in two steps; the coarse registration and the fine registration. The registration is split in these two steps when the application of one registration step is not enough to bring the point clouds in a perfect match. Particularly, the coarse registration refers to the estimation of an imperfect matching position of two overlapping point clouds [Byun et al., 2017]. Figure 2.2a illustrates the idea of two objects being coarsely matched. A fine registration is applied on the coarsely matched point clouds as a final and improved step (Figure 2.2b). It is a re-estimation of the transformation parameters so as to enhance inaccurate or coarse estimates of low quality [King et al., 2005].

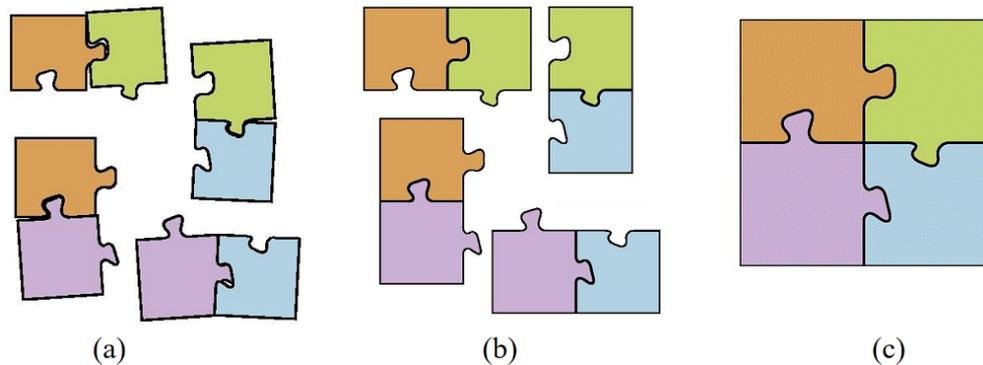


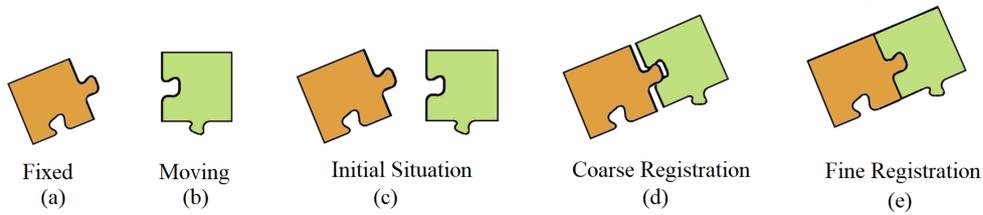
Figure 2.2: Puzzle pieces are used for visualization purposes. (a) It illustrates the result of a coarse pairwise registration, thus the puzzles are not completely matched. (b) It illustrates the result of a fine registration, thus the puzzles are perfectly matched. (c) It illustrates the result of a global registration, where each matched pair is aligned with the other local pair matches. (Modified figures from [Theiler et al., 2015])

In this project, a course registration is not really necessary. Since in this project we deal with data that accompanied with location information, even though the positions of the point clouds might have large errors, there is still some information about their initial position. If the project was dealing with images for which no positioning information was available, then the coarse registration would be a necessity.

### 2.1.2 Fixed & moving point clouds

Researchers usually consider that there is one scan which defines the coordinate system of the matching result when dealing with local registration. In other words, a scan which is free from positional errors. This is termed as fixed [Kang et al., 2009], target [Huang et al., 2008], model [Besl and McKay, 1992] or reference [Pomerleau et al., 2013]. The other scan needs to be transformed into the coordinate system of the fixed scan and is termed as moving [Kang et al., 2009], source [Huang et al., 2008], data scan [Besl and McKay, 1992] or reading [Pomerleau et al., 2013]. In this project the terms fixed and moving are used to describe the point clouds in a pair. Figure 2.3a depicts a fixed scan and Figure 2.3b depicts a moving scan. For the better understanding of having a fixed and moving scan, the initial positions (Figure 2.3c) of Figure 2.3a and Figure 2.3b, their coarse registration (Figure 2.3d) and their fine registration (Figure 2.3e) are illustrated.

Nevertheless, for the registration of mobile laser scanning data none of the point clouds in a pair can be used as the fixed scan. It is possible that a point cloud that overlaps with some other point clouds has minimum positioning error (order of millimeter). However, even if that is the case, there is no knowledge which point cloud among some overlapping point clouds has minimum positioning errors. Despite this lack of information, the relative transformation parameters would be



**Figure 2.3:** Puzzle pieces are used for visualization purposes. (c) The initial positions of a fixed scan (a) and a moving scan (b). (d) A course registration is applied where the moving scan is almost matched onto the fixed scan. (e) A fine registration is applied to improve the coarse registration result.

the same even if one point cloud or another is considered to be the fixed one. Only the directions (positive, negative) of the displacements will have to change. Therefore, the selection of the fixed scan does not influence the relative registration, but it influences the global registration. In Chapter 7 a recommendation is given on how the relative translations, can be used to facilitate the global registration. A method is suggested with which the relative offset between a [point-cloud pair](#) could be distributed between the overlapping scans. The suggested method depends on the positioning quality of the overlapping point clouds.

## 2.2 GLOBAL POINT CLOUD REGISTRATION

Global registration is also known as multiview registration [Pulli, 1999]. This term describes the process of aligning, in a common reference system, all the point cloud scans that have been acquired along the trajectory of a vehicle performing [MLS](#) [Nüchter et al., 2005]. The implementation of this step implies the alignment of the all the local matchings together (Figure 2.2c). Namely, all the matched scans resulted from the relative registration need to be adjusted so that they can correctly co-exist and form an integrated 3D point cloud model. To achieve that, global registration methods distribute the registration error over all the scans [Krishnan et al., 2007].

## 2.3 QUALITY OF LOCAL POINT CLOUD REGISTRATION

Quality is a measure of the degree to which a product meets the user's demands [Teunissen et al., 2006]. A stand alone estimated solution is not enough. Each estimation must be accompanied with a quality measure. In this project a quality indication is needed for the estimated transformation parameters. To define the quality of the local point cloud registrations, theoretical knowledge from the Adjustment and Testing theory for [geodetic observations](#) mainly developed by Baarda [1968], studied thoroughly by Teunissen et al. [2006] and applied by Sweco Nederland B.V. [2016] is used. The quality of a measure in the 'Adjustment theory' is described by the aspects of reliability and precision [Teunissen et al., 2006]. In order to provide definitions about the reliability and precision of a local transformation solution, reliability and precision are initially generally described. Then definitions of these terms are given for geodetic networks. The proposed method adjusts the theory described in this section to the problem of local registration of [LiDAR](#) data in Section 4.7.

**THE RELIABILITY OF A SYSTEM OR A PRODUCT** Reliability has been studied by many researchers from different disciplines. The reliability of a product or a service as explained by Teunissen et al. [2006] refers to: 1) the degree to which errors and

anomalies, or misspecifications of the product in general can be traced, and 2) the degree of sensitivity of the final results to errors, anomalies and misspecifications that remain undetected. A more general definition is given by Grous [2013], where reliability refers to the discipline that analyzes failure of a product. According to Gnedenko et al. [1999] reliability is the prediction of the estimation's confidence limits. Neubeck [2004] defines the reliability of a system as the detection of its pluses and minuses to verify the system's functionality.

**THE PRECISION OF A SYSTEM OR A PRODUCT** Precision refers to the variability of the results expected when a process of a product is observed many times under similar circumstances [Teunissen et al., 2006]. The variability is caused by random errors in the process and in the measurements.

**GEODETIC OBSERVATIONS VS LIDAR DATA** The Adjustment and Testing theory which is used to compute reliability and precision has been extensively studied for traditional geodetic observations, but not for modern observations such as LiDAR data. The main difference between the two is that the LiDAR data is enormously dense. For example millions of points can be found in a square tile of  $250m^2$  (as shown in Figure 1.7). In contrast, a few hundreds of geodetic observations could be collected for an area of  $250m^2$ . Nevertheless, both LiDAR and geodetic observations include spatial information. Thus the reliability and precision defined for geodetic observations could be used to provide adjusted definitions for the purposes of the local registration of LiDAR data. The definitions of the reliability and precision of geodetic observation follows. These definitions are adjusted for point clouds local registration and presented in Section 4.7.

**THE RELIABILITY OF GEODETIC OBSERVATIONS** The reliability is subdivided into internal and external reliability. Internal reliability is the size of the smallest possible observation error, where a large size indicates a weakly checked observation. The internal reliability can be explained with the Redundancy Number of the observations [Sweco Nederland B.V., 2016]. To be more precise, the redundancy is achieved when redundant measurements are retrieved, while the measurements obey to some mathematical rules [Teunissen, 2000]. For example, instead of measuring only two angles of a triangle and calculating the third, measuring all the angles. The external reliability is used as a measure to determine the influence of a possible undetected error in the observations that affect the adjusted coordinates [Sweco Nederland B.V., 2016]. Adjustment of the coordinates is necessary when multiple, but unequal measurements exist.

**THE PRECISION OF GEODETIC OBSERVATIONS** Precision describes the degree of closeness of the observations when repetition of observations is possible [Sweco Nederland B.V., 2016]. A highly precise observation indicates that a measurement is repeated many times. It is likely possible that repeating a measurement will not always give the same answer.

## 2.4 POINT CLOUDS PROCESSING

This section presents techniques used for LiDAR point clouds processing. Firstly, the cases where a 3D point constitutes an outlier with respect to the local registration are explained. Next, a method for rejecting outlying points based on their density is explained. Lastly, the basic concept for the computation of the normal vector on 3D points is described.

### 2.4.1 Rejection of outlying points

**DEFINITION OF OUTLIERS IN THE POINT CLOUDS** A 3D point cloud registration method may converge to a solution driven by false matches due to outliers or noise presented in the point clouds. An outlying point with respect to the relative mobile LiDAR point clouds registration is a point that does not belong to the topography of the area [Matkan et al., 2014]. Outlying points can be clusters of points or single points [Pang, 2011]. The following four cases indicate when 3D points represent outliers which can possibly cause registration results of degraded quality.

Points that represent moving objects, such as cars, people [Matkan et al., 2014] or suspended objects at high altitude, such as birds and smoke, are considered outliers. These constitute clusters of outliers and describe specific objects. Moving objects may influence negatively the registration of overlapping point cloud pairs (Figure 2.4). For example, if a car is parked when a scene is initially recorded, but during the second recording the car is not there anymore, or is moving, or has parked in another spot then the two point clouds do not share common information. Thus, the registration could be negatively influenced. If the moving objects captured at the overlapping point clouds are the same position, then they do not constitute a problem for the local registration. In contrast, they will represent extra common information which may facilitate the matching. An example of moving objects that can be scanned at different times but remain at the same position are parked cars.

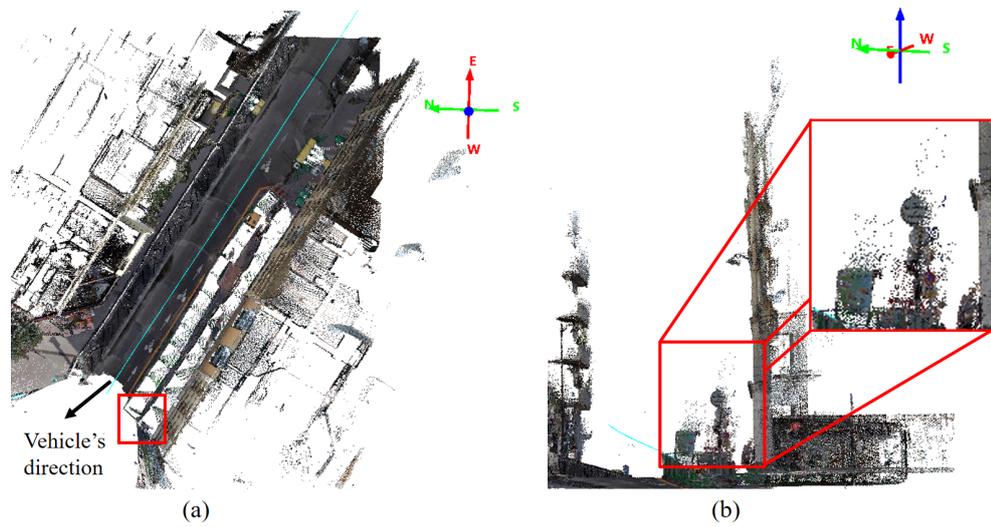


**Figure 2.4:** a) Point cloud square tile of  $250m^2$  in which moving objects (cars) have been scanned. These are circled in red. b) Point cloud square tile of  $250m^2$  representing the same area as (a) in which moving objects have not been scanned.

In cases where the laser beam hits the boundary of an object, but that object blocks other objects, then artificial points are scanned between the two objects (Figure 2.5). These artificial points are produced between two surfaces because the footprint of the laser beam is not a single point but a geometrical Gaussian ellipse [Sotoodeh, 2006] [Naidu and Fisher, 1991]. When the laser beam hits a boundary of an object which occludes another object, the elliptical footprint of the laser beam is split and artificial points are measured. The irradiance of the artificial points equals a weighted average of the irradiance from the two involved surfaces; the occluded and the one that occludes. These points constitute noise in the point clouds.

Additionally, in cases of surfaces that return very high or very low reflection values such as black objects, glasses and smooth metal surfaces [Sotoodeh, 2006] noise is collected. Namely, recording such surfaces result in biases in the distance measurements and thus, to increased noise [Beraldin, 2004] in the point clouds.

When the angle formed by the Z axis of the scanner and the laser beam is significantly small (i.e smaller than  $45^\circ$ ), then the object is hit under a low angle. In cases were objects are hit under a low angle, the laser beam is firstly deflected onto neighbor objects and then reflected back to the receiver. This causes longer

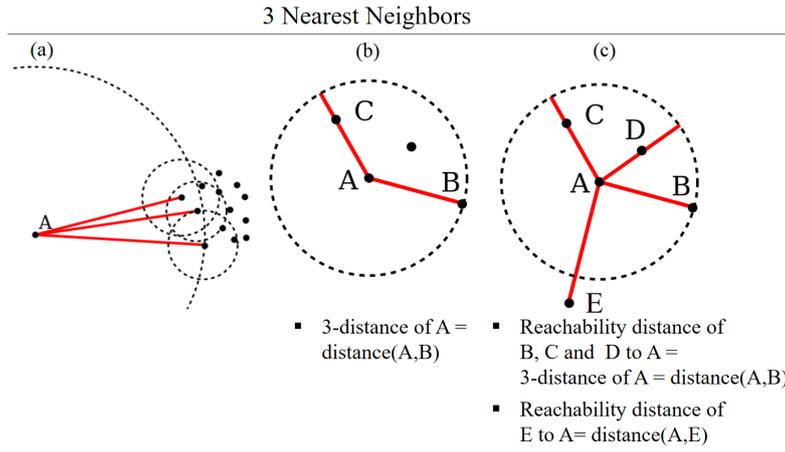


**Figure 2.5:** a) Point cloud square tile of  $250m^2$  viewed from the top. The light blue line indicates the direction of the recording vehicle. The red square includes the area of interest which is explained in (b). b) Side view of the point cloud tile of (a). Three objects are included in the magnification. A short object, a high object and a wall. From the perspective of the laser beam direction the wall is occluded by the high object and the high object is occluded by the short object. Therefore noise is captured in between the objects.

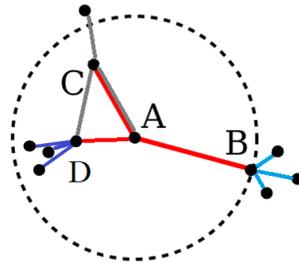
traveling time of the laser beams [Sotoodeh, 2006]. As a result, noise is generated in the point clouds.

**AN OUTLIER REJECTION METHOD** The Local Outlier Factor (**LOF**) is a property that indicates the degree to which a point is isolated from its neighborhood [Breunig et al., 2000]. It can be used as an outlier rejection method. It is calculated by comparing the point's local density with the local densities of its neighboring points (Figure 2.6a). The locality is given by the nearest neighbors of a point, and the distance to the neighboring points is used to estimate the density. To compute the **LOF** of each point it is necessary first to detect the  $k$ -Nearest Neighbors (**k-NN**) of each point. These are the  $k$  (number of) points that are closest to the observed point. Then the 'reachability distance' is considered as it is needed for the computation of the **LOF** of each point. The reachability distance of a point is the maximum distance between the actual distance of a neighboring point to the observed point, and the  $k$ -distance of the observed point. The  $k$ -distance metric of an observed point is the distance within which at least  $k$  points can be found (Figure 2.6b). The reachability distance of all the points that do not constitute a nearest neighbor of the examined point is their actual distance from it. The reachability distance of all the **k-NN** is the  $k$ -distance of the examined point (Figure 2.6c). The reachability distance is introduced in order to reduce the statistical fluctuations for all the neighboring points of the observed point. In such a way, more stable (**LOF**) results within clusters of points are produced.

To compute the **LOF** of each point, it is necessary first to compute the Local Reachability Density (**lrd**) of each point. The Local Reachability Density of a point equals to the inverse of the average reachability distances with which the point can be reached from its neighbors. A more detailed explanation about the computation of the **lrd** of a point is given through the Figure 2.7. The general Equation for the computation of the **lrd** of a point is shown in the equation 2.1. The Equation 2.2 shows how the equation 2.1 is modified specifically for the example illustrated in Figure 2.7.



**Figure 2.6:** The following illustrations take into account 3-nearest neighbors for each point of interest. (a) Representation of the basic idea of the Local Outlier Factor: the comparison of the local density of a point with the local densities of its neighboring points. The local density of A is much lower than that of its neighbors. (b) The  $k$ -distance of A with  $k=3$  is the distance within which the 3 nearest neighbors can be found. The  $k$ -distance equals the distance of A and B, as B is the 3rd neighbor of A, namely the 3rd closest neighbor to A. (c) The reachability distance of B, C and D with respect to A is the same and equals to the  $k$ -distance with  $k=3$ . In contrast, E is not one of the 3 nearest neighbors of A, and thus its reachability distance from A is the actual distance of D and A. (Figures (a) and (b) from Wikipedia. Figure (c) modified from Wikipedia).



**Figure 2.7:** Example for the calculation of the Local Reachability Density of point A with  $k=3$  nearest neighbors. The red lines connect A with its 3 nearest neighbors which are B, C and D. The light blue lines connect B with its 3 nearest neighbors. The gray lines connect C with its 3 nearest neighbors. The dark blue lines connect D with its 3 nearest neighbors. The point A is used as the observed point for which the `lrd` needs to be computed. The reachability distances at which A can be reached from its neighbors are needed for the `lrd` calculation. The reachability distance at which A can be reached from its neighbor C is the  $k$ -distance (3-distance) of C, because A is one of the 3 neighbors of C. The 3-distance of C is coincidentally the distance from C to A, as A is the 3rd furthest neighbor of C. The reachability distance at which A can be reached from its neighbor B is the actual distance from A to B and not the 3-distance of B, since A is not one of the neighbors of B. The same is valid for the reachability distance at which A can be reached from its neighbor D. The Local Reachability Density of A is equal to the inverse of the average reachability distances with which A can be reached from its neighbors (Equation 2.2).

$$lrd(p_j) = \frac{1}{\sum_{i=1}^k rd(p_j, p_i) / k} \quad (2.1)$$

where  $lrd(p_j)$ : local reachability density of a point  $j$

$k$ : number of nearest neighbours

$rd(p_j, p_i)$ : reachability distance of point  $j$  from a neighboring point  $i$

$$\frac{1}{(\text{distance}(A, C) + \text{distance}(A, B) + \text{distance}(A, D))/3} \quad (2.2)$$

where  $\text{distance}(A, C)$ : the actual distance between the observed point A and the neighboring point C  
 $\text{distance}(A, B)$ : the actual distance between the observed point A and the neighboring point B  
 $\text{distance}(A, D)$ : the actual distance between the observed point A and the neighboring point D  
3: the number of the nearest neighbors

Next, the **LOF** can be computed by comparing the **lrd** of the observed point and the **lrd** of the neighbors. Particularly, the **LOF** equals the average of the **lrd** of the observed point neighbors divided by the observed point's own **lrd**.

$$LOF(p_j)_k = \frac{\sum_{i=1}^k \text{lrd}(p_i)/k}{\text{lrd}(p_j)} \quad (2.3)$$

where  $LOF(p_j)_k$ : The Local Outlier Factor of a point j based on its k nearest neighbors  
 $\text{lrd}(p_i)$ : local reachability density of a point i  
 $\text{lrd}(p_j)$ : local reachability density of a point j

Breunig et al. [2000] prove that a **LOF** value close to 1 represents an object with density comparable to its neighbors, and thus not an outlier. Objects with **LOF** value below 1 indicate a dense cluster, thus an inlier point. In contrast, values significantly larger than 1 indicate outliers. The **LOF** method does not make the assumption that being an outlier is a binary property. Since each point gets a (**LOF**) score which describes the degree to which a point constitutes an outlier. This gives the opportunity to specify some thresholds to define the inliers and the outliers.

#### 2.4.2 Normal vectors

The normal vector of a point is the vector which is perpendicular to the tangent plane fitted on an observed point and its neighbouring points. Therefore, the estimation of the normal vector includes the search for each point's neighbors. The neighboring points can be found for example based on a fixed distance from the observed point or based on a specific amount of points considered as neighbors (Figure 2.8a). In order to find how the points in each neighborhood are spread, the local variation of a neighbourhood is taken into account. An example of the local variation in a neighbourhood of points with a covariance ellipse is visualized in Figure 2.8b. The local spread of the points is described with 3 vectors as shown in Figure 2.8b. The vector with smallest vector value determines the direction of the least variation, and is the normal vector of the observed point (Figure 2.8b). The normal vector is perpendicular to the direction of most variations. The Figure 2.8c shows the normal vectors of a triangular mesh for a better understanding.

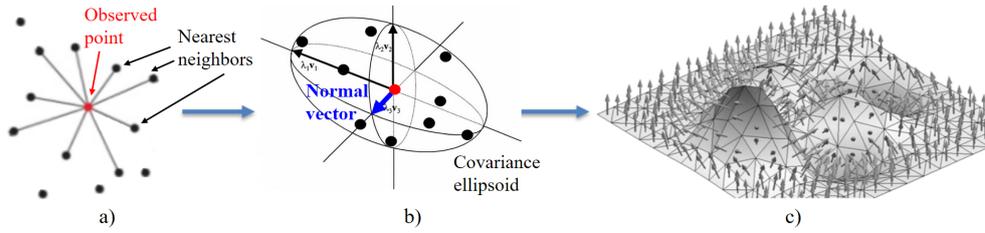


Figure 2.8: a) The red point shows the observed point and the black points are its neighbors. b) The covariance ellipsoid that shows the local variation of an observed point and its neighboring points. The blue vector is the normal vector of the observed point and its neighbourhood. c) Visualization of the normal vectors of a triangular mesh. The normals are perpendicular to each planar triangular face.

### 2.4.3 The Principal Component analysis method

The Principal Component Analysis (PCA) is a statistical approach used, among others, for the computation of the normal vectors of the surfaces on 3D points. To get insight regarding the variations of the points from their centre of gravity the covariance matrix of the neighbourhood of each point is computed. The equation of the covariance matrix for 3D points is shown in Equation 2.4. The values inputted in the covariance matrix are the covariances of all the possible combinations between the variables X, Y and Z. Equation 2.5 illustrates an example of how the covariance between two variables is computed. The next step is the computation of the eigenvectors and the eigenvalues of each covariance matrix. Then eigenvectors and eigenvalues show the dispersion of the neighbors in the 3 directions around the examined point (Figure 2.8b). The eigenvector of which the corresponding eigenvalue is the least among the 3 eigenvalues indicates the least variation. That eigenvector is considered the normal vector of the plane that fits best to the neighborhood of points [Smith, 2002]. The normal of the fitted plane can be used also as the normal vector of the examined point.

$$C = \begin{pmatrix} cov(x, x) & cov(x, y) & cov(x, z) \\ cov(y, x) & cov(y, y) & cov(y, z) \\ cov(z, x) & cov(z, y) & cov(z, z) \end{pmatrix} \quad (2.4)$$

$$cov(x, y) = \sum_{i=1}^N [(X_i - \bar{X})(Y_i - \bar{Y})] / (N - 1) \quad (2.5)$$

where  $X_i, Y_i$ : the coordinates of an observed point  
 $\bar{X}, \bar{Y}$ : the mean X and mean Y values of the points in  
the neighbourhood of  $point_i$   
 $N$ : the amount of points in a neighbourhood

## 2.5 IMAGE PROCESSING

This subsection describes techniques that can be applied on images. These techniques are the image gradient, the Sobel operator and the Gaussian smoothing is given. Also the method template matching which is suitable for registering images is analyzed.

### 2.5.1 Image gradient

The gradient of an image shows how the values of an image change in the horizontal and vertical direction. The change is represented by a direction and a magnitude. To retrieve such information it is necessary to compute the first derivatives of the values of an image in the X and Y directions and combine those into a vector [Jacobs, 2005].

As the gradient has a direction and a magnitude, the gradient is usually encoded as a vector. An example of a gradient vector applied to a point of an image is shown in Figure 2.9. The left part of the Figure 2.9 shows a 2D image and the direction of the change of the values at a specific point. The right part of the Figure 2.9 shows the same as the left part but in 3D. Namely, at the right part the brightness values of the image are imagined as height values. The direction uphill is the direction of the gradient, pointing in the direction of the largest possible increase of the values [Flores-Mangas, 2014].

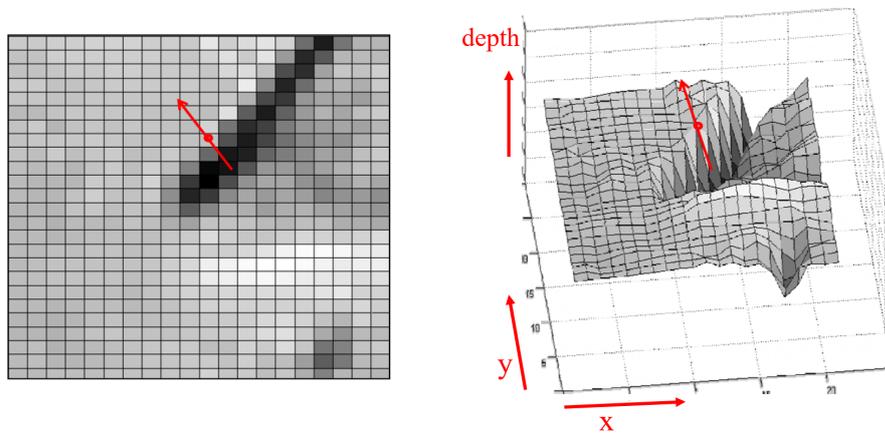
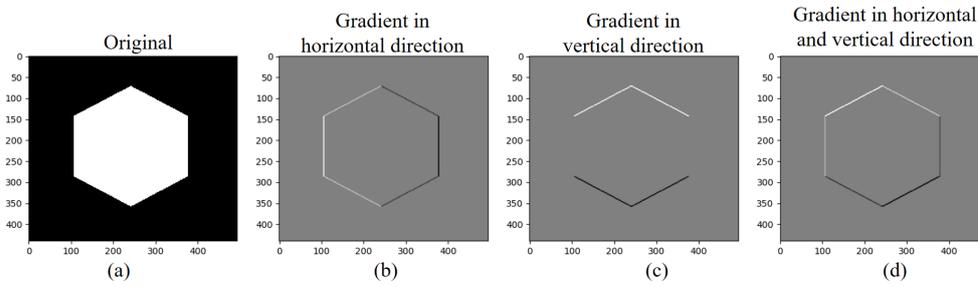


Figure 2.9: A 2D image and its corresponding representation in 3D. It is shown at a specific point of the image how the brightness values change. In the 3D representation, the direction of the gradient is the direction uphill, pointing in the direction of largest possible value increase [Flores-Mangas, 2014]. (Figures from [Flores-Mangas, 2014]).

Figure 2.10a shows a simple black and white image. The gradient in the horizontal direction is computed for the Figure 2.10a and the result is illustrated in Figure 2.10b. The constant gray values inside the hexagon and at the background indicate that there is no change in the values of the Figure 2.10a. The bright values in the horizontal gradient (Figure 2.10b) indicate change from low (dark) values to a high (bright) values, thus change from left to right. In contrast, the dark values indicate change from (high) bright values to (low) dark values. The original image is scanned from the left to the right to compute the gradient. Thus, firstly the dark values of the Figure 2.10a are encountered and then the bright resulting to the brighter gradient values at the left side of the hexagon. Subsequently, dark values are encountered first (inside the hexagon) in the Figure 2.10a and then bright resulting to dark gradient values at the right side of the hexagon. Figure 2.10c shows the rate of change in the vertical direction of the values in Figure 2.10a. Brighter gradient values are shown at the upper part of the hexagon as the image is scanned from up to down encountering first low pixel values and then high. Darker gradient values are shown at the lower part of the hexagon as the image is scanned from up to down encountering first the high pixel values inside the hexagon and then the high values in the background. The left and right sides of the hexagon are gray as there is no change in the values. The Figure 2.10d shows a combination of the Figures 2.10b and 2.10c. It illustrates the gradient in both the horizontal and vertical direction of the Figure 2.10a.



**Figure 2.10:** a) A black and white image. b) The result of the gradient in the horizontal direction applied on the image shown in (a). c) The result of the gradient in the vertical direction applied on the image shown in (a). d) The result of the gradient in the horizontal and vertical direction applied on the image shown in (a).

### 2.5.2 Sobel Operator

A Sobel operator could be used for the computation of the first derivative of an image. A Sobel operator does not actually calculate the first derivative of an image. However, it calculates an approximation of it by fitting a polynomial with discrete values [Bradski and Kaehler, 2008]. To apply a Sobel operator a convolution operation must be used. Convolution in the spatial domain is the process of using a matrix, or a kernel as it usually called, which slides over the values of an image and modifies them by performing a specific calculation. At each overlay, each element of the kernel is multiplied with the corresponding element on the image. Then all the results of the multiplications are summed up. The output is placed in the new image at the pixel as that of the center of the kernel<sup>1</sup>.

The Sobel operator can be performed for kernels of any size [Bradski and Kaehler, 2008]. There is one kernel to compute the horizontal changes and one to compute the vertical changes. Here an example of symmetric 3X3 kernel used to apply the Sobel operator in X and Y direction is given in Equations 2.6 and 2.7 correspondingly. In order to calculate the magnitude or in other words the rate of change in one image the Equation 2.8 is used [Lemmens, 1992].

$$Gx_{Sobel} = G * \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.6)$$

$$Gy_{Sobel} = G * \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (2.7)$$

where  $G$ : the image on which the operator is applied

$Gx_{Sobel}$ : the output of the Sobel operator in the X direction

$Gy_{Sobel}$ : the output of the Sobel operator in the Y direction

$$GM_{Sobel} = \sqrt{Gx_{Sobel}^2 + Gy_{Sobel}^2} \quad (2.8)$$

<sup>1</sup> The convolution in the spatial domain is usually implemented as a multiplication in the frequency domain through the Fast Fourier Transform, because it is a significantly faster than the convolution in the spatial domain.

### 2.5.3 Gaussian Smoothing

Smoothing is another convolution operation often applied in order to reduce noise on images. An example of a smoothing filter is the Gaussian Smoothing, which is used to remove Gaussian noise from images. The  $3 \times 3$  kernel of the Gaussian Smoothing is given in Equation 2.9 [Lemmens, 1992].

$$SMOGAU = G * \begin{pmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{pmatrix} \quad (2.9)$$

where  $G$ : the image on which the operator is applied  
 $SMOGAU$ : the output of the GAUSSian SMOothing

### 2.5.4 Image registration

Image registration constitutes the process of matching images that contain common visual information Gaidhane et al. [2014], but possibly retrieved at different epochs, from different viewpoints, and by different instruments [Norollah et al., 2012]. The Figures 2.11a and 2.11b illustrate an example of two images retrieved from different viewpoints. The Figure 2.11c illustrates the registration result of the Figures 2.11a and 2.11b. It can be seen that the common visual information of the Figures 2.11a and 2.11b is matched.

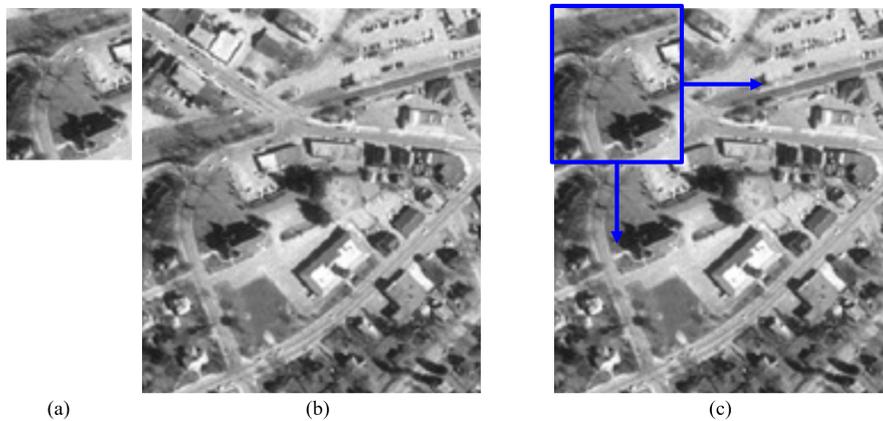


Figure 2.11: (a) and (b) Images containing common visual information, but retrieved from different viewpoints. (c) The registration result of (a) and (b). (Figures: MATLAB Tutorials).

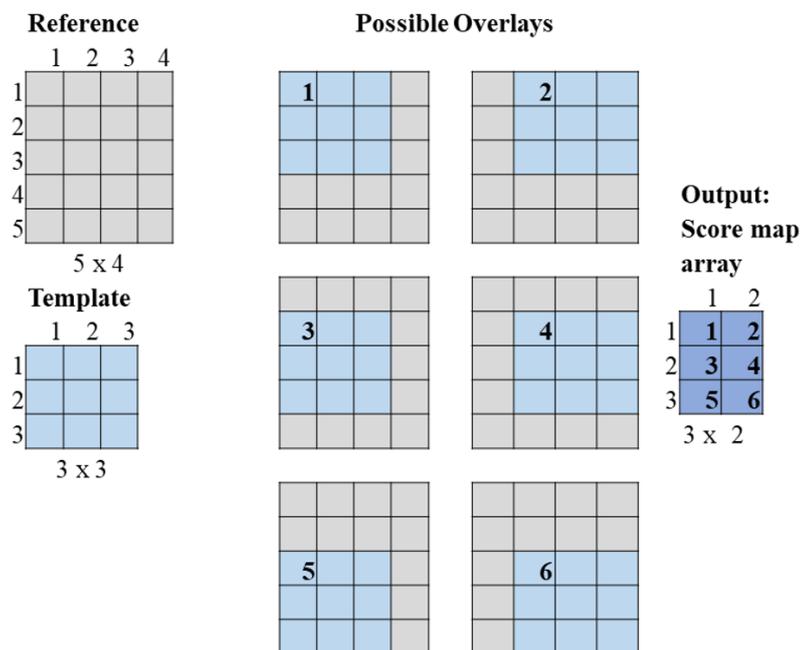
### 2.5.5 Template matching

A template matching technique can be used for image registration. Template matching is the process of determining the presence and the location of a template image inside a reference image [Zhang et al., 2009]. A template image usually corresponds to an image smaller than the reference image (Figures 2.12a and 2.12b). This difference in the size of the two images is necessary to create some search space between the two images. Since the requirement is to detect the location of the template image in the reference image, the template image is the one which should be smaller. The concept of template matching is that the template image slides over every possible location of the reference image, pixel by pixel, from left to right and from up to down (Figure 2.12c). At every location, a degree of similarity between the two images is calculated, or in other words a metric that shows how good or bad the match is. Thus, the output of the template matching is an array containing the similarity scores calculated at every possible overlay between the 2 images. This array is called the score map, as it contains the similarity scores of mapped locations

between the template and the reference image. Its size equals to  $(W - w + 1) \times (H - h + 1)$  in pixels, where  $W$  and  $H$  the width and the height of the reference image, and  $w$  and  $h$  the width and the height of the template image, correspondingly. The size of the output score map array is based on the number of possible positions that the template could be placed on the reference image. Figure 2.13 illustrates a relevant example.



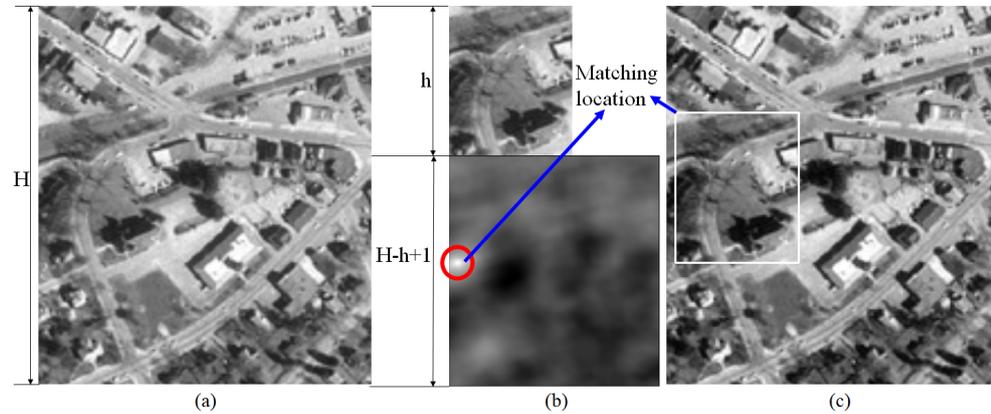
**Figure 2.12:** (a) An example of a template image. (b) An example of a reference image. (c) The location of the template image shown in (a) is searched into the reference image shown in (b). The template is sliced over the reference so as to compare them and find the best matching location.



**Figure 2.13:** A reference image where its *Width* equals to 4 and its *Height* equals to 5. A template image where both its *width* and *height* equal to 3. Next to the reference image the 6 possible overlays between the reference and the template image are illustrated. The 6 possible overlays indicate where the template can be superimposed on the reference without having pixels 'hanging out' of the reference image. Therefore, the resulted score map array has only 6 cells constructed from width equal to 2 ( $Width - width + 1 = 2$ ) and height equal to 3 ( $Height - height + 1 = 3$ ).

The location (simply the image coordinates) of the highest similarity score in the score map defines the matching location of the template image into the reference image. Particularly, the matching location refers to the upper left corner of the

template image in the reference image. The Figure 2.14a illustrates a reference image. The Figure 2.14b illustrates at its upper part a template image of which the position must be found in the reference image. The lower part of Figure 2.14b shows the output of a template matching technique, namely the score map array. The score map array obtains color according to the similarity values calculated at every overlay between the template and the reference image. Thus the score map array is also called score map image. The brighter the pixel, the higher the similarity. In Figure 2.14c the matching location is searched in the reference image to spot the location of the template image.



**Figure 2.14:** a) A reference image. b) The upper part shows a template image. The lower part shows the resulted score map array from a template matching. The red circle indicates the pixel with the highest similarity, or in other words the matching location of the top left corner of the template image onto the reference image. c) The matching location of the template image is detected on the reference image. The white outline indicates the boundaries of the template image on the reference image.

# 3

## RELATED WORK

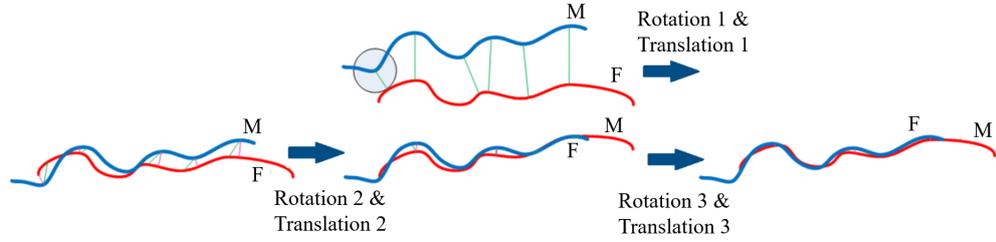
This chapter is split in two main sections, the 3D Local Point Cloud Registration (Section 3.1) and the 2D Local Point Cloud Registration (Section 3.2). The aim of the Section 3D Local Point Cloud Registration is to show which concepts and methods are followed by other researchers to solve relative registration of point clouds in 3D. Thorough explanation on the most commonly used algorithm for point cloud registration, the Iterative Closest Point (ICP), is given (Subsection 3.1.1). Additionally, some variants of ICP are discussed (Subsection 3.1.2), quality assessment methods are examined (Subsection 3.1.3) and outlier rejection methods are reviewed (Subsection 3.1.4). Conclusions upon the suitability of registering point clouds in 3D are also provided (Subsection 3.1.5). The aim of the Section 3.2, 2D Local Point Cloud Registration is to provide insight on techniques that could be used to develop an image-based local point cloud registration method. The section contains information about existing image-based local point cloud registration approaches (Subsection 3.2.1) and types of images that could be generated from the point clouds (Subsection 3.2.2). Moreover methods used to obtain subpixel accuracy values from imagery, or in other words from discrete values, are investigated (Subsection 3.2.3). Lastly, conclusions upon the suitability of registering point clouds in 2D with existing approaches are discussed (Subsection 3.2.4).

### 3.1 3D LOCAL POINT CLOUD REGISTRATION

#### 3.1.1 Point clouds local registration with ICP

In this subsection, the concept of the ICP algorithm and some limitations of it are explained, as this algorithm is often used for point cloud registration in 3D. The main ICP algorithm was initially developed by [Besl and McKay, 1992]. Multiple modified versions of it have been developed to deliver improved results [Byun et al., 2017].

As stated in Section 2.1.2 (Fixed & moving point clouds), the notions fixed and moving are used to describe the point clouds in a point-cloud pair. 'Fixed'  $F$  is the point cloud that is considered to have correct coordinate system. 'Moving'  $M$  is the point cloud that has to be moved to match the fixed one. The ICP algorithm computes repetitively the transformation parameters by reforming point associations between the fixed and the moving scan (Figure 3.1). The algorithm iterates until the Mean Square Error (MSE) of the distances between the correspondences is sufficiently small, or the MSE difference between two consequent iterations is sufficiently small, or if the maximum allowed amount of iterations is achieved. When one of these conditions is satisfied the motion that matches  $M$  to  $F$  is calculated and applied to  $M$ .



**Figure 3.1:** The Iterative Closest Point algorithm that matches a moving scan to a fixed scan by iteratively minimizing the distance between the computed correspondences.

Equation 3.1 indicates that according to the determined point correspondences, the sum of the squares of the residuals between the points in a fixed (**F**) scan and their rotated and translated corresponding points in a moving **M** (scan), must be minimum.

$$(R, t) = \min \left( \sum_{i=1}^N \| \mathbf{F}_i - R * \mathbf{M}_i - t \|^2 \right) \quad (3.1)$$

where  $R$ : Rotation parameter

$t$ : translation parameter

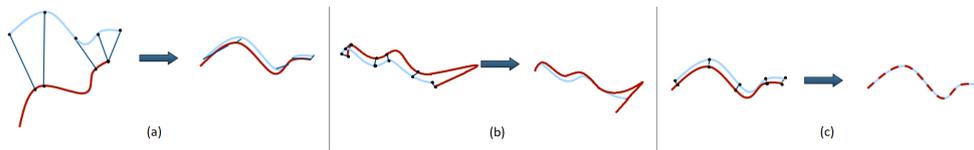
$F_i$ : A point of the fixed scan

$M_i$ : A point of the moving scan

**ICP** implementations may vary according to the features registered. For example, the main **ICP** registers point to point features [Besl and McKay, 1992], a method of Chen and Medioni [1992] registers points to planes, while a method of Segal et al. [2009] registers planes to planes etc. **ICP** point to point features perform better than the point to plane **ICP** if there are quadratic or polynomial geometric (curved) structures in the scene [Bellekens et al., 2014]. **ICP** point-to-plane [Chen and Medioni, 1992] provide better results than the point-to-point if the structures in the environment are characterized by many straight lines [Pomerleau et al., 2013]. In cases where a lot of noise is observed in the point clouds, the **ICP** plane-to-plane would outperform the **ICP** point-to-plane [Bellekens et al., 2014].

### 3.1.2 Point clouds local Registration with ICP variants

Variants of **ICP** provide optimal local registration results when the initial positions of the overlapping point clouds are close to the registration solution [Shetty, 2017], [Cartwright, W. et al., 2009], [Wang C. et al., 2014], [Trucco et al., 1999]. In contrast, as explained in the section 2.1.1, the initial positions of the overlapping point clouds in the case of mobile laser scanning may be far away from each other. Additionally, **ICP** methods are negatively influenced when many incorrect point correspondences are estimated because a corresponding point does not exist in the other point cloud. This occurs due to the partial overlap between the overlapping point clouds [Pomerleau et al., 2013]. Therefore, **ICP** provides results of high quality when all the points or many points in one scan have correspondence in the other [Trucco et al., 1999]. Having points in one scan without correspondence in the other is quite common in the case of **MLS** data as it was shown in the Figures 1.7 and 1.8 on pages 7 and 8 respectively. Lastly, the need for correspondences establishment in 3D requires an expensive nearest neighbors search [Sanchez et al., 2017]. Thus, the execution time of the registration with **ICP** is considerably high [Langerwisch and Wagner, 2010]. An overview of the **ICP** algorithm's limitations is provided with the illustrations of Figure 3.2.



**Figure 3.2:** The Iterative Closest Point (ICP) algorithm converges to a false local minimum Mean Square Error when: (a) the initial alignment of the two scans is not sufficiently accurate and (b) the two scans do not contain the same information or, simply put, do not fully overlap. (c) ICP algorithm provides appropriate results when the two scans fully overlap (or share considerably large overlap) and their initial positions do not differ significantly from the final matched positions.

A description of some algorithms that are based on ICP follows. These algorithms utilize the main concept of ICP, but are modified in order to deal with its limitations. Usually the computation of point correspondences is alternated in order to detect and filter out incorrect correspondences. These constitute incorrect associations between points of two scans. Incorrect correspondences often occur when points that exist in one scan do not exist in the other scan of a [point-cloud pair](#). A point cloud scan might have less points than another scan, due to the vehicle's long distance from the scene during the recording.

The purpose of describing a few ICP variants is to stress the avails and limitations of using an ICP-based method for local point cloud registration. These are summarized in Section 3.1.5.

**ITERATIVE CLOSEST COMPATIBLE POINT** The Iterative Closest Compatible Point (ICCP) algorithm developed by [Godin et al. \[1994\]](#) differs from ICP as ICCP seeks for the correspondences between the points under a constraint. This constraint is a compatibility measure based on the intensity of the points. Points are compared based on their intensity, for each point, the points that have similar intensity value are determined. Subsequently, the point with the minimum distance is chosen as the corresponding point. The search space is reduced since the corresponding point is searched only among the points which have similar intensity. Thus, the most computationally expensive operation of ICP, the detection of correspondences, is improved. Despite this, the compatible points with regard to the intensity are recomputed at each iteration which is a costly operation. ICCP, like ICP, performs suitably when most of the points in the one point cloud have a correspondence in the other.

**ROBUST ITERATIVE CLOSEST POINT** The difference of the Robust Iterative Closest Point (RICP) [[Trucco et al., 1999](#)] and the ICP is that a Least Median Squares (LMedS) method is applied [[Rousseeuw, 1984](#)] to eliminate the incorrect correspondences. A point from each point cloud set is randomly selected. Then, the registration transformation between the two points is calculated and applied to the moving scan's point. Then, their residual is computed. This process is repeated until all the potential registrations are evaluated and the solution that returns the LMedS value is chosen. The residuals that are larger than a threshold given by the user are rejected. Just as the ICP [[Besl and McKay, 1992](#)], so too the RICP depends on the initial positions of the point clouds. Nevertheless, it provides better results than the ICP if there is a high presence of incorrect correspondences. In other words, if the overlap between the overlapping point clouds is small the results have good quality [Trucco et al. \[1999\]](#).

**TRIMMED ITERATIVE CLOSEST POINT** The trimmed-ICP algorithm, developed by [Chetverikov et al. \[2002\]](#), is based on the Least Trimmed Squares (LTS) method which was introduced by [Rousseeuw \[1984\]](#). This method focuses on the distances between determined corresponding points in point cloud pairs. It sorts the square

distances and minimizes their sum by iteratively excluding a number of extreme values [Chetverikov et al., 2002]. Namely, it trims the largest distances to prevent influencing the point cloud registration. The optimal property of *LTS* relates to its high *Breakdown Point* which is equal to 50%. A *Breakdown Point* determines the degree of robustness of an estimate in the presence of outliers. It measures the amount of outliers an estimate can handle before giving an incorrect estimate [Hampel, 1971]. The high *Breakdown Point* of *LTS* is the advantage of this method. It indicates that the Trimmed-*ICP* can handle highly contaminated data. This refers to incorrect correspondences [Cizek and Visek, 2000]. A limitation of this method is that it assumes a fixed overlap of scans [Pomerleau et al., 2013]. The algorithm requires knowledge of the overlap percentage in order to know how many point correspondences to trim. If this method was used for the local registration of mobile laser scanned point clouds, the overlap of every *point-cloud pair* should be calculated, leading to additional computation time. Sorting the determined point correspondences by the distance between them in order to remove incorrect correspondences, adds to the computational load [Padia and Pears, 2017].

**ITERATIVE CLOSEST POINT USING INVARIANT FEATURES** The algorithm Iterative Closest Point using Invariant Features was developed by Sharp et al. [2002]. It deals with the improvement of the correspondences selection by extracting features invariant to 3D rigid motion from the point clouds. These features do not alternate when arbitrary rotations and translations are applied. Such a feature is the curvature, which is the amount by which a geometric object deviates from being a flat plane, or the amount by which a curve object deviates from being a line. Point locations are related to the extracted features to determine the correspondences in the point clouds that overlap. The benefit of the Iterative Closest Point using Invariant Features (*ICPIF*) algorithm is that fewer iterations are required than in the *ICP* to converge to a solution. However, the *ICPIF* constructs correct point correspondences when the point clouds are free from noise.

### 3.1.3 Local registration quality

In this Subsection techniques used for the quality assessment of local point cloud registration methods are examined.

The quality of relatively registered data is evaluated in a research work by assessing whether the overlapping region after alignment represents the same physical surface [Huber and Hebert, 2003]. Some measures of surface consistency were used such as the Euclidean distance between corresponding points of two overlapping point clouds and the angle between the normal vectors of corresponding points. A surface is considered consistent when these two measures are less than a threshold. In another work, similar to the Euclidean distance, a mean-square error *MSE* was computed after the detection of point correspondences [King et al., 2005]. The matching solution is accepted when the *MSE* is close to the approximate noise of the sensor. In addition, a non-randomness score is used to detect the random alignments or mis-alignments due to repetitive structures. An example of a repetitive structure could be a row of windows in a building. This non-randomness score relates to the output of the matching of distinctive structures detected on the two point clouds. Namely, the matching is implemented two times, but with two different methods. If the transformation parameters are close to the first registration estimation, then this accounts as a proof that the firstly estimated solution is non-random.

### 3.1.4 Outlier rejection to minimize influence on registration

In this Subsection, some methods for rejecting outlying points in 3D are discussed. It is necessary to find how the outliers could be rejected to decrease their effect on the registration results.

The definition of outliers with respect to point clouds registration is extensively described in Subsection 2.4.1. Briefly, outliers can be either moving objects or noise. The outliers that represent moving objects could be rejected from the point clouds by developing an algorithm for moving objects detection. However, such a step would be out of the scope of this project. Therefore, the research focuses only on the noise in the point clouds which might decrease the quality of the registration.

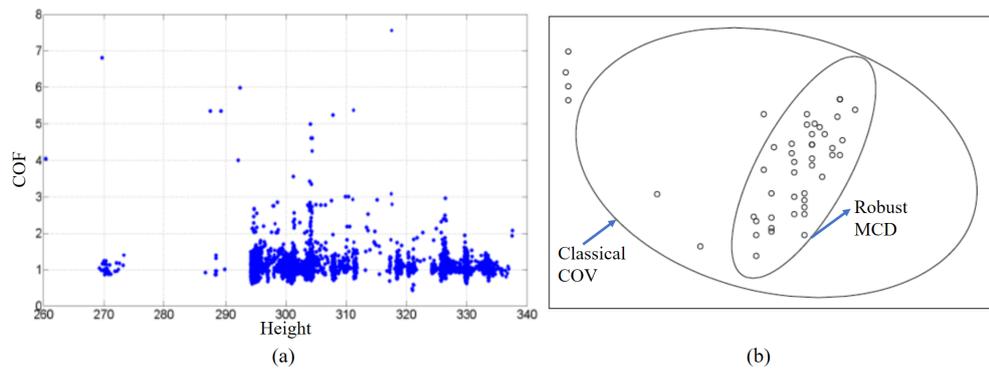
Matkan et al. [2014] use a cross-validation technique and specifically the method leave-one-out to reject outliers based on the height variations from the surrounding points. This cross-validation technique considers every data point of a LiDAR point cloud once as the testing sample and excludes it from the rest of the data-set. Then, an interpolation method is applied to the elevation component of the neighboring points of the observed point to find its interpolated height. The error of the point is calculated by subtracting its actual height from the predicted height, given from the interpolation. This process is then iterated to all the points in a point cloud. Next, a threshold is determined based on the maximum height error value, which is more robust than choosing a random number. Then, the points that gave an error value higher than the threshold are rejected. All the previous steps are repeated and the algorithm terminates when the maximum height error value among the errors from all the points is higher than the maximum error of the previous iteration.

The chosen cross validation method is less biased than other validation methods as every sample is considered once as test-data. However, it is very computationally expensive since every point is taken as a testing sample. Also the discussed method rejects outliers only based on the relative height between the test point and its neighbors. The spatial relationship of the points is neglected. Particularly the method does not take into account how isolated or nearby a point is from other points.

A multi-attribute model is used by Pang [2011] to detect 3D outliers. The elevation of each point is taken into account and a Connectivity Outlier Factor (COF). The COF indicates how isolated a point is from its neighbors. These two attributes are considered for each point. Then, the attributes are combined to form a 2D space as shown in Figure 3.3a. A Minimum Covariance Determinant (MCD) is used to detect the outliers. This constitutes a robust estimator of location and scatter [Rousseeuw, 1985] for outlier rejection in comparison with the classical covariance. The classical covariance is also an estimator of location and scatter, but highly affected by outliers [Pang, 2011]. Figure 3.3b visualizes with covariance ellipses the comparison between the classical covariance and the robust MCD for outlier rejection. Further explanation of the MCD concept follows.

The MCD of some neighboring 3D points refers to the minimum number of points that are not outliers. The concept of MCD is applied by using a robust Mahalanobis distance. The Robust Mahalanobis distance is applied on neighborhoods of points in the formed 2D-attribute space. The Robust Mahalanobis distance is calculated for each point to determine how far the point is from a tolerance ellipse. The tolerance ellipse is formulated by considering the mean, the variance and the covariance of the elevation and COF attributes of the points. If the Robust Mahalanobis distance of the point is higher than the threshold, then the point is considered an outlier. The threshold is the formulated tolerance ellipse, visualized in Figure 3.3b.

This method detects individual and clusters of outliers. However, the relationship of the two attributes is not very well integrated in the model which causes the detection of more outliers than really exist [Pang, 2011]. Figure 3.3a is observed to describe a relevant example. If a point has a bit higher COF value than



**Figure 3.3:** (a) An example of a 2D space based on the elevation and the Connectivity Outlier Factor of each point in a point cloud. b) According to the given point observations, the result of the classical covariance is illustrated with the large ellipse. According to the given point observations, the result of the robust covariance is illustrated with the small ellipse.

its surrounding points, but its elevation is comparable to the rest of the points then the point might be considered as outlier because of the slightly different COF value. Nonetheless, the removal of more outliers could increase the efficiency of the registration process, however there should be a balance. Considering many points as outliers while they are not, may lead to registration results of degraded quality.

### 3.1.5 Conclusions

This subsection contains some conclusions drawn after the analysis of the literature concerning the point clouds registration and the outlier rejection methods in 3D.

#### WHY TO REGISTER IN 3D BY USING ICP-BASED METHODS

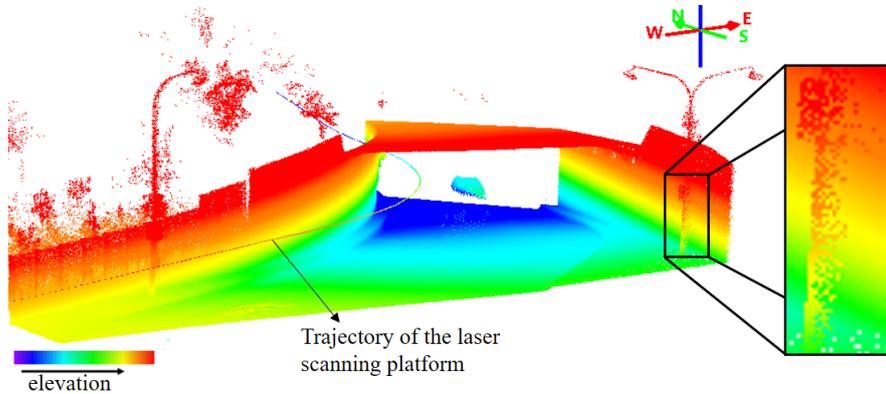
- By registering with ICP in 3D or with a variant of ICP in 3D, or with any other 3D registration method, the matching will have the accuracy with which the points have been collected. LiDAR point clouds are usually collected with high accuracy, thus the registration in 3D will have high accuracy.

#### WHY NOT TO REGISTER IN 3D BY USING ICP-BASED METHODS

- ICP-based algorithms are computationally expensive due to the need for search and detection of point correspondences in point-cloud pairs.
- ICP-based algorithms require good estimations of the initial positions of the overlapping point clouds. Else the algorithm minimizes the MSE by associating wrong points.
- ICP-based algorithms require that the overlapping point clouds have all or a large number of the corresponding points in both scans. Nevertheless, this is not the case with the mobile scanned point clouds. The amount of collected points may vary according to the distance and orientation of the vehicle with respect to the surroundings.

**OUTLIER REJECTION IN 3D** A factor which can contribute to the determination of the isolation of a point from its neighboring points, such as the COF used from Pang [2011], is considered of high importance. The determination of neighbourhoods of points can facilitate the description and comparison of the spatial relationship of the points. However, it is challenging to define how big a neighbourhood should be so as to allow for detection of all the outliers in a point cloud set. For example in a square point cloud tile of  $250m^2$  there might be sections where a small neighbourhood must be used to detect the outliers. In other sections of the  $250m^2$

area possibly large neighbourhoods must be used to detect the outliers. Therefore, the spatial relationship should be combined with an extra attribute, as performed in the research of Pang [2011] by using the elevation of the points. Nevertheless, the elevation may not be the best option for judging if a point belongs to the topology of an area or not. Let us take as an example the points representing a light pole in Figure 3.4.



**Figure 3.4:** Example of a square point cloud tile of  $250m^2$  where the color gradations represent the elevation values. The magnified part illustrates a part of a light pole in the scene. The emphasis is on the noise that appears close to the pole. The noise exists due to the occlusion of the vertical structure from the pole with respect to the laser beam. In such a case, the elevation of an inlier point and the elevation of an outlier point at a certain horizontal plane will be the same. Thus, the elevation will not be a representative factor to describe an outlier. In contrast, it can be seen that the outlier points are very sparse in comparison to the points that actually represent the pole. Therefore, density of a spatial neighbourhood could be used as another attribute to assess whether a point is an outlier or not.

The method of Matkan et al. [2014] which employs a cross-validation technique to compare the actual elevation of each point with a predicted elevation value is considered a very expensive method. Besides this, it takes into account only the elevation of the points, and as explained this might not provide very representative results.

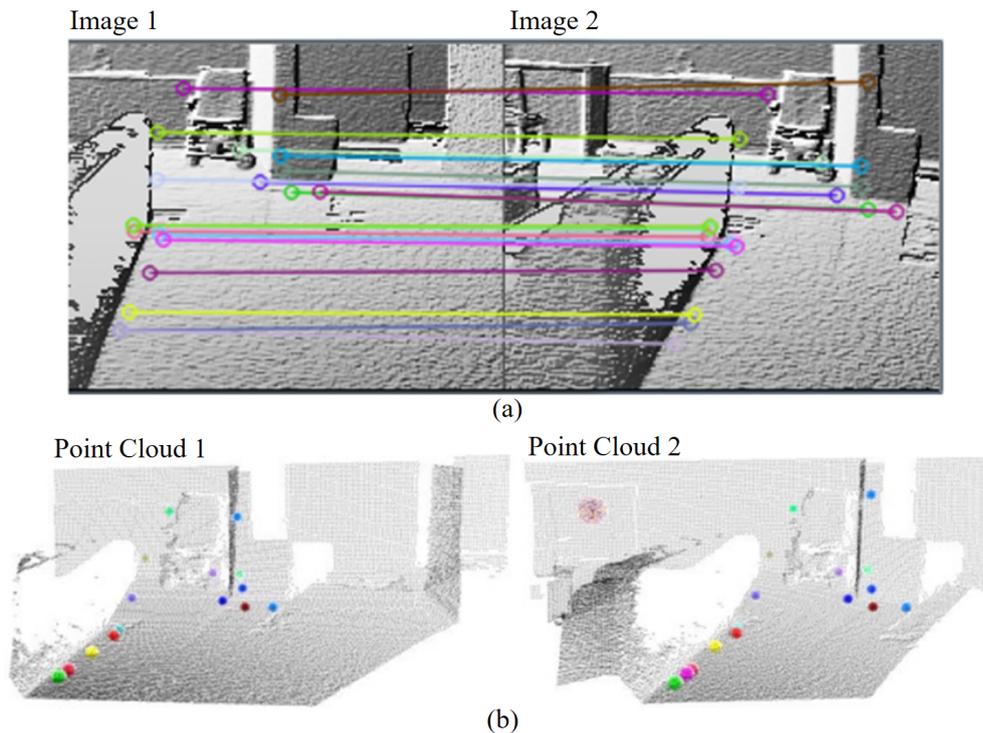
To summarize, the aim is to use a method that takes at least into account the spatial relationship of the points so as to provide information about the isolation of the points with respect to their neighbors. Also, the points' density should be used to judge if a point is an outlier or not. Further attributes, such as the intensity and the elevation of the points could be combined to provide more robust results.

## 3.2 2D LOCAL POINT CLOUD REGISTRATION

The aim of the 2D Local Point Cloud Registration section is to provide insight into techniques that could be used to develop an image based local registration method. Other image-based local point cloud registration approaches are discussed, and types of images that could be generated from the point clouds are described. Additionally, methods for retrieving sub-pixel accuracy registration results are discussed. Lastly, conclusions upon the suitability of registering point clouds in 2D with existing approaches are discussed.

### 3.2.1 Image-based local point cloud registration

Bearing angle images (which are described in Section 3.2.2) are generated from 3D point clouds [Lin et al., 2017]. Then, a feature based matching method is used to find corresponding pixels between an image pair. The (2D) pixel correspondences are filtered to avoid including incorrect correspondences in the solution. The Figure 3.5a illustrates an example of two bearing angle images with common visual information, along with the pixel correspondences extracted from the feature-based matching. The (2D) pixel correspondences are converted in 3D (point correspondences). The Figure 3.5b illustrates the corresponding 2D correspondences in 3D, namely on the point cloud pair. The point correspondences are used in a least squares approximation to derive the transformation parameters, which is basically the main step of ICP. The sum of the squares of the residuals of the distances between the corresponding points is minimized in order to obtain the optimal transformation parameters.



**Figure 3.5:** (a) Two bearing angle images that highlight the edges of the objects in the diagonal direction, extracted from a two point clouds. In addition, the correspondences found in the images by using a feature descriptor are shown. (b) The correspondences found in 2D are converted and presented in the 3D space. (Figures from [Lin et al., 2017]).

The benefit of using a feature based matching method is that an initial alignment of the point clouds is not needed. Lin et al. [2017] compare their algorithm with the generalized-ICP, which constitutes a plane to plane matching Segal et al. [2009]. They

show that the computation cost is significantly less than *ICP*, due to the 2D matching. However, the precision is not better than that of generalized-*ICP*, because sometimes incorrect correspondences are included.

Furthermore, a method developed by [Liang et al., 2017] generates perspective intensity images<sup>1</sup> by constructing central projections of terrestrial point clouds. After the construction of the perspective intensity images, corner points are detected in the imagery and used as corresponding points to match the images. The advantage of this work is that building structures become very distinct on the perspective intensity images, which facilitates the registration process [Liang et al., 2017]. Despite this, one should generate perspective images from many viewpoints to achieve complete representation of a 3D point cloud set.

### 3.2.2 From 3D point clouds to 2D imagery

Some techniques for converting 3D data to 2D have been explored, discussed and compared in this section.

**DEPTH IMAGES** Point cloud data can be converted to depth images that convey per pixel depth information [Chmelar et al., 2016]. The depths are distance values between each point and a view point of the point cloud, thus their computation is simple. Depth maps are useful for distinguishing which objects are closer to the view point and which are further away. Figure 3.6a illustrates an intensity image of an indoor place in RGB. Figure 3.6b illustrates the same indoor place as a depth image in grayscale. The closer the object is to the camera point, the brighter the object is in the depth image. In contrast, the further away from the camera the darker the object is in the depth image.

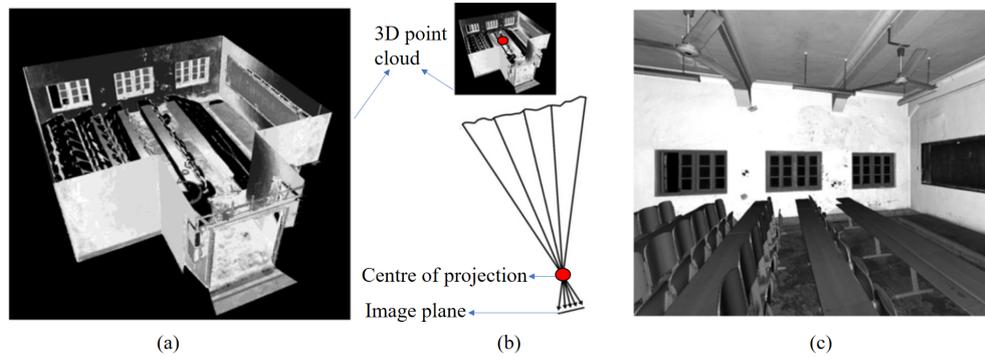


Figure 3.6: a) An image with intensity values in RGB colours. b) The depth image of Figure (a) in grayscale. (Figures from NCTech [2018].)

**PERSPECTIVE IMAGES** 3D data can be converted to perspective intensity images [Liang et al., 2017]. The generation of this type of images requires the use of the collinearity equation which is commonly used in Photogrammetry. This equation is used in order calculate the planar coordinates of the 3D points while performing a central projection. A relevant example is visualized in Figure 3.7.

**BEARING ANGLE IMAGES** Bearing angle images have been proposed by [Scaramuzza et al., 2007]. They can be used to stress discontinuities of depth and changes of directions in the point clouds. The bearing angle images have the ability to stress the aforementioned image details at specific directions. For example, by highlighting the depth discontinuities only at the horizontal or vertical or diagonal direction.

<sup>1</sup> The perspective images are described in the upcoming Section (3.2.2).

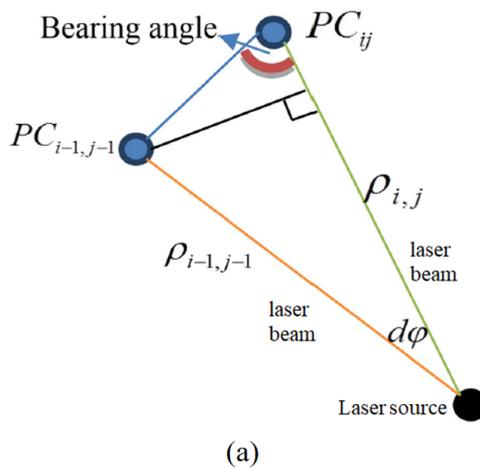


**Figure 3.7:** (a) A screen-shot of a point cloud. (b) A central projection on the point cloud of (a). The centre of projection, illustrated with red color, is placed inside the scene of the point cloud as shown in the miniature. (c) The retrieved perspective intensity image in grayscale projected from the point visualized in (b). (Modified Figures from Liang et al. [2017].)

These images illustrate the angle between the laser beam and the vector that joins 2 consecutive measurements of points (Figure 3.8). To define the consecutive points it is necessary first to arrange the point measurements in a 2D array where the entries of the array are ordered based on the direction of the laser beam. According to the direction in which is desired to stress the depth discontinuities the consecutive points are detected. The bearing angle value is computed for each point as a function of the depth value of a point, the depth value of its adjacent measurement point, the distance between the two and the angle formed by the consecutive points and the laser beam. Specifically, the Equation 3.2 indicates how the bearing angle for each point is calculated.

$$BA_i = \arccos \frac{\rho_i - \rho_{i-1} * \cos d\phi}{\sqrt{\rho_i^2 - \rho_{i-1}^2 - \rho_i * \rho_{i-1} * \cos d\phi}} \quad (3.2)$$

where  $\rho_i$ : the depth value of the point  $i$  in the selected direction of the depth array  
 $\rho_{i-1}$ : the depth value of the point  $i-1$  in the selected direction of the depth array  
 $d\phi$ : the angle of the laser beam in the desired direction



**Figure 3.8:** The blue circles illustrate two consecutively scanned 3D points. The bearing angle is the angle which is formulated by the vector which connects the two points and the secondly reflected laser beam (Figure from Lin et al. [2017]).

The Figure 3.9 illustrates an example of four bearing angle images where the consecutive points were taken into account in four different directions.

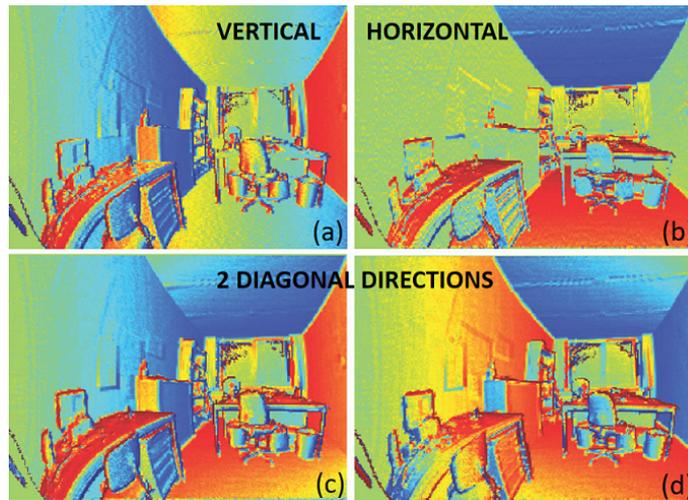


Figure 3.9: (a) Bearing angle images in the vertical direction, (b) the horizontal direction, (c) and (d) in the diagonal directions ( $-45^\circ$ ,  $+45^\circ$ ). The color shade added from blue to red is proportional to the bearing angle value (Figures from [Scaramuzza et al., 2007]).

**HISTOGRAM BINNING** The histogram binning approaches are discussed by Blomley et al. [2014]. Additionally, in an unpublished work [van Someren, 2016] and Christodoulou [2017] histogram binning methods are developed. Specifically, a set of 2 histograms per point cloud are created, one for the X and Y values (the xy-histogram) and one for the Z values (z-histogram). Particularly, for the creation of the xy-histogram the point clouds are projected on a horizontal plane viewed from the top. Next, a 2D grid is created on the horizontal plane, and the points are binned into the cells according to their X and Y coordinates. The value of each 2D grid cell (pixel) represents the number of points that fall within the cell. The Figure illustrates the creation of an xy-histogram. For the Z-histogram, the point cloud is considered as a 3D volume sliced horizontally. Specifically, the point clouds are 'viewed' from the side, then sliced horizontally and each created horizontal cell corresponds to a grid cell (pixel) in the Z-histogram. The value of each pixel represents the number of points that fall within the corresponding 1D grid cell. The Figure illustrates the creation of an xy-histogram.

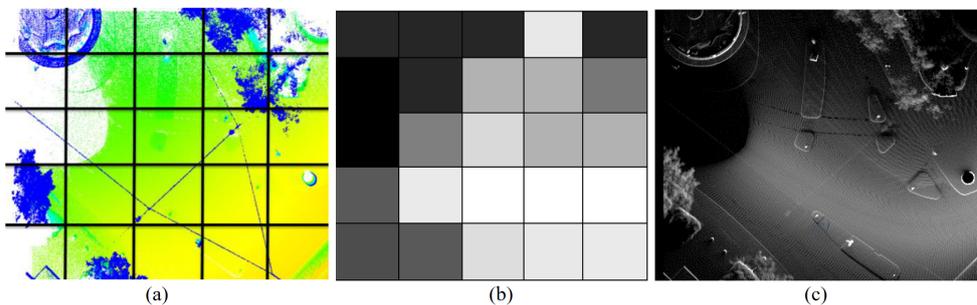
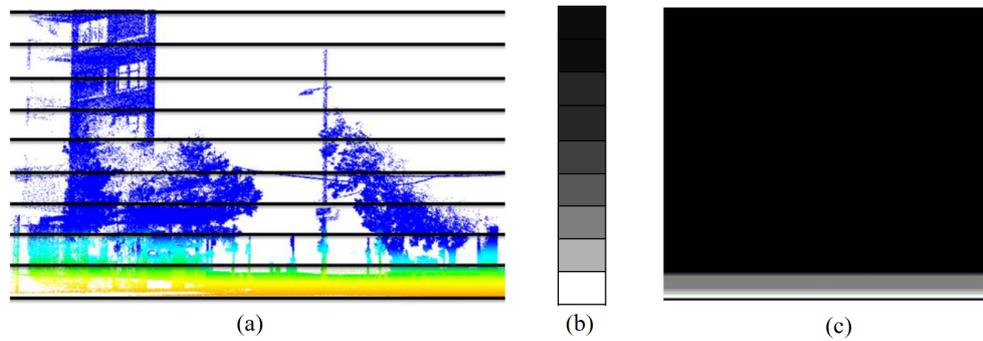


Figure 3.10: A visual example of the xy-histogram creation. (a) The point cloud is projected to a horizontal plane viewed from the top and sliced in 2D grid cells. (b) Each grid cell of (a) corresponds to a pixel in the xy-histogram image. Each pixel carries the frequency value of the points that fall within the 2D cell. The less the points, the darker the pixel. The examples (a) and (b) consider very large pixels for visualization purposes. (c) It illustrates the xy-histogram image of the point cloud (a) with a small grid cell size (0.1m x 0.1 m).

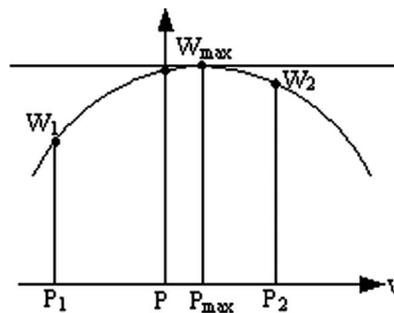


**Figure 3.11:** A visual example of the z-histogram creation. (a) The point cloud is considered as a 3D volume, which is sliced horizontally. (b) Each horizontal cell of (a) corresponds to a pixel in the Z-histogram image. Each pixel carries the frequency value of the points that fall within the corresponding 1D cell. The less the points, the darker the pixel. The example (a) and (b) considers very large pixels for visualization purposes. (c) It illustrates the z-histogram image of the point cloud (a) with a small grid cell size (0.1m x 0.1 m). The picture is stretched horizontally visualization purposes.

### 3.2.3 Sub-pixel accuracy

As mentioned in Section 1.2 (Objectives & research question) the relative transformation parameters between point clouds will be calculated by developing an image-based technique. As a result, the parameters can have at maximum the accuracy of the images' grid cell size. Techniques to retrieve values of sub-pixel accuracy, or in other words, accuracy higher than that of the pixel, are explored.

A second order polynomial is fitted on the values of some pixels of interest [Zhang et al., 2009] to retrieve sub-pixel values.  $P_1$ ,  $P$  and  $P_2$  of Figure 3.12 represent three neighbouring pixels. It can be seen that when a polynomial is fitted on the three pixels, values of sub-pixel accuracy are generated. For instance, information is acquired about the highest possible value between the area that the three pixels cover. This is the value  $W_{max}$  of the point  $P_{max}$  and not the value  $W$  of pixel  $P$ , as one would think by examining the discrete pixel values. In another work, the fitting of a 1D Gaussian function is discussed [Naidu and Fisher, 1991] and applied with the same concept as explained for the polynomial fitting.



**Figure 3.12:**  $P_1$ ,  $P$ ,  $P_2$  represent the discrete locations of three pixels.  $W_1$ ,  $W$  and  $W_2$  are the corresponding pixel values. The discrete pixel values are interpolated with a second order polynomial to retrieve sub-pixel accuracy values.

### 3.2.4 Conclusions

**IMAGE-BASED LOCAL POINT CLOUD REGISTRATION** Both image-based point cloud registration methods described in Section 3.2.1, extract and match distinct features such as corner points on the generated image pairs. Therefore, the registration in both approaches is based on the determined correspondences. The approach with which bearing angle images are created and matched has worse precision than that of generalized-ICP due to the inclusion of wrong point correspondences [Lin et al., 2017]. Thus, although it is a 2D-based method for point cloud registration, it has similar limitations as ICP. In both methods the incorrect correspondences is the reason for which the registration results are negatively influenced. In contrast, the method in which perspective images are created and matched has accuracy of millimeter level Liang et al. [2017]. However, that method was only tested with three point cloud pairs. Therefore, the conclusions cannot be representative for various data-sets.

Although the bearing angle images have the ability to highlight the discontinuities of the depth of the objects, they must be computed for every desired direction. For example, if it is desired to stress the image details in the horizontal direction and the vertical direction then two different computations must be implemented. However, the reason why the bearing angle images were introduced was because there was a requirement to stress the image details in different directions Scaramuzza et al. [2007]. Otherwise, the normal images could be generated as they can also illustrate the orientations of the surfaces Scaramuzza et al. [2007]. If the normal vector is split in its three directions (X, Y and Z), then it is possible to highlight the surfaces' orientations in the different directions without performing additional computations for each direction. In contrast, as mentioned, this is required for the construction of the bearing angle images.

Furthermore, the registration method in which perspective images are created was tested only on a few point clouds Liang et al. [2017]. Therefore, the location of the centre of projection could be manually chosen to ensure that the resulted images will have overlap. Some overlap, or in other words common visual information, is necessary for the images registration. However, for mobile laser scanned point clouds retrieved from different positions the selection of the centre of projection it is more complex. The centre of projections must be selected automatically because many point clouds are collected and need to be matched. Also, the automatically selected centre of projection must ensure that there is overlap between two created images.

More information about the suitability of different image types for image registration is given in Section 4.4.1. That section explains why specific images are preferred and how they are created, as part of the proposed point cloud registration method.



# 4

## IMAGE-BASED POINT CLOUD LOCAL REGISTRATION APPROACH

This chapter presents the proposed image-based point cloud local registration approach from a conceptual point of view.

### 4.1 MOTIVATION

The key motivation behind the proposed method follows from the limitations of the techniques that already exist with respect to mobile point clouds. The 3D ICP-based point cloud methods require that the initial positions of the point-cloud pairs are close to the final solution. However, as explained (Section 3.1.2), when it comes to MLS data, in many cases the initial offset between the overlapping point clouds is large. Also, ICP-based methods have high performance when the overlap between the point clouds that capture the same scene is complete. With point clouds retrieved from MLS processes it is not common to have complete overlap. Large overlap is possible when both scans have been captured from short distance, however also small overlap is possible when at least one scan in a pair has been captured from long distance. Furthermore, the ICP variants require extensive and expensive search for correspondences between the point clouds that overlap. This is not ideal for a project that deals with MLS data, as an immense amount of 3D points may be scanned in a small time period. Consequently, the method must have the potential to converge to solutions in a considerably short amount of time. From this perspective, a 2D-based method would be more suitable than a method that registers the point clouds directly, in 3D. This is believed because if the point clouds are converted to images and the images are matched, then the method will be less dependant on the number of 3D points. Instead, the method will be more dependant on the number of the pixels in the generated imagery. There will be certainly less pixels in the imagery than points in the points clouds, when point clouds are projected on 2D, requiring less computation.

In Chapter 3, the existing methods for point cloud registration in 3D and 2D were explained and judged. The method which converts the point clouds to bearing angle images [Lin et al., 2017] (Section 3.2.1), delivers results with lower quality than the Generalized-ICP which matches planes to planes (constructed from the points). The reason for this is the determination of incorrect point correspondences between the point clouds pairs. Therefore, the requirement is to develop a method that either constructs robust point correspondences or avoids completely the concept of the point correspondences. Furthermore, neither the method using bearing angle images [Lin et al., 2017] nor the method using perspective images [Liang et al., 2017] deal with mobile point clouds. That indicates that the methods do not deal with big amounts of data, neither with large offsets between the point clouds nor with small overlaps between the point clouds. Therefore, it is not known how the existing methods perform in those cases.

As explained, it is believed that an image-based point cloud registration method is promising and thus, an image-based method is developed. The aim of the developed method is to overcome the limitations faced by the ICP-based methods and the existed image-based methods, while focusing the design of the method on mobile laser scanned point clouds. Then, it will be possible to answer the question

of this research, namely: *To which extent is it possible to automatically, reliably, precisely and efficiently align mobile laser scanning data relatively, using an image-based technique?*

The following sections present the algorithm developed to answer the research question of this thesis. Many steps of the algorithm are based on the theory presented in Chapter 2. Implementation details are found in Chapter 5. For every decision the motivation and reasoning is explained. Section 4.2 provides a global overview of the whole approach, while the following sections describe the different steps in the approach in more detail.

## 4.2 METHOD OVERVIEW

Figure 4.1 illustrates an overview of the developed method. Point cloud tiles are the input to the designed algorithm. The initial three steps of the method are part of the pre-processing. A method based on the density of the points' neighbourhoods is used to reject outlying points, particularly noise. Subsequently, for every point cloud tile the normal vectors of the surfaces are computed on the points. The last step of the pre-processing is the construction of **point-cloud pairs**; pairs of overlapping point cloud tiles. The Pre-processing steps are described in Section 4.3. Thereafter 2D projections are created from the 3D point cloud tiles. The reduction of the dimensions is applied with various techniques which are described in Section 4.4. The following step which is explained in Section 4.5 constitutes a multi-registration approach, as various techniques are used to convert the input 3D data to 2D. Therefore redundant 2D registration solutions are estimated. Section 4.6 indicates how the redundant 2D solutions are converted to 3D. Section 4.7 provides information concerning the manner with which the low quality results among the redundant solutions are discarded to estimate the optimal translation parameters. In section 4.8, the technique used to retrieve transformation parameters of sub-pixel accuracy is discussed. This technique is applied on the optimal translation solutions retrieved from the main image-based registration method in order to increase the translations' accuracy. Lastly, the 2D sub-pixel accuracy translation parameters are converted back to the 3D space coordinates.

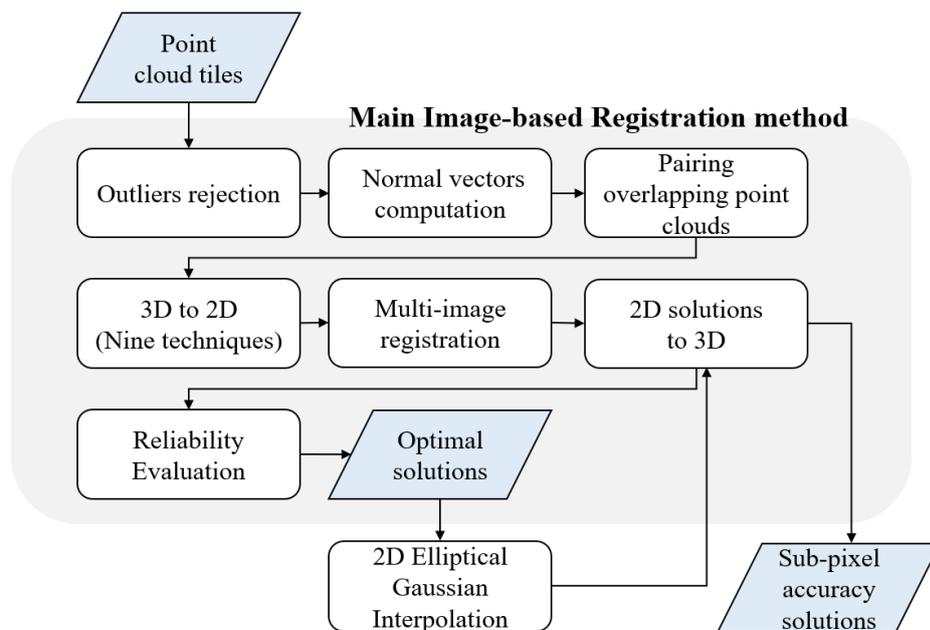


Figure 4.1: Overview of the proposed image-based point cloud registration method.

## 4.3 PRE-PROCESSING

This section discusses the initial steps of the developed method; the rejection of the outliers, the computation of the normal vectors and the creation of point cloud pairs. The developed method is applied on point cloud tiles, which are also mentioned as [point-cloud sets](#).

### 4.3.1 Outlier rejection

Outlier points in the point clouds are defined in Section 2.4.1. The outliers are split in two categories, the moving objects and the noise. As stated in Section 3.1.4, the development of a method for the detection and rejection of moving objects is outside of the scope of this project. Thus, the focus lays on how to reject the noise from the point clouds so that its influence on the registration results is limited.

**REASONING** The concepts behind the proposed approach are based on existing methods as discussed in Section 3.1.5.

- It is necessary to consider a factor that takes into account the spatial relationship of the points. In other words, each point should be compared to their neighbouring points. For example, as shown in the Figures 4.2a and 4.2b, the points in the neighborhoods that represent the noise are more scattered than the points in the neighbourhoods of inliers.
- The density of the points' neighbourhoods could be used as a factor to detect outliers. However, the density of neighbourhoods of points that were far away from the recording system will also be low. A related example is illustrated in Figure 4.2b in the upper red circle. It is possible that points that are not outliers will be considered outliers with a method that compares the density of points. However, this would only be undesired if the registration results of the developed method are negatively influenced. If the results are not becoming worse while more points are rejected, then this could actually work as a benefit, as the method will have to process fewer points. That could lead to decreased computation time of the algorithm.
- The elevation of a point (used in [Pang, 2011] and Matkan et al. [2014]) might not be the best parameter to judge if a point belongs to the topology of an area or not. As it was shown in Figure 4.2c the elevation of the outliers does not differ from the elevation of the inliers.

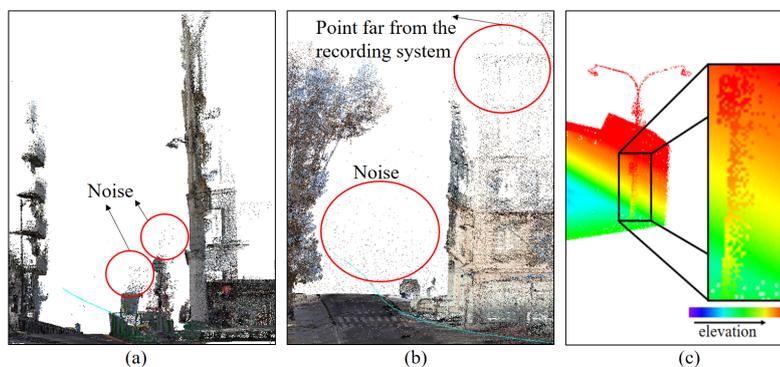


Figure 4.2: a) Noise in the point clouds, where the points are very scattered. b) Same as (a) but it also illustrates scattered points that are not noise. c) The elevation of the outliers can be the same as the elevation of the inliers.

Consequently, a method that computes the local density of the points is used for the reduction of noise. With this method, the degree to which an object is isolated from its surrounding neighborhoods is utilized. That degree of isolation is known as the local outlier factor (LOF). The computation of the LOF is given by comparing the local density of an object to the local densities of its neighbors [Breunig et al., 2000]. The locality is given by the nearest neighbors of a point. The advantage of this approach over others is that it gives the ability to determine which points are outliers and which not by configuring the threshold based on the computed LOF scores of the points.

**METHOD** The concept of the LOF method and its design details are thoroughly explained in the theoretical background in Section 2.4.1, page 14. These are used in order to calculate the LOF of the points. Implementation details are given in Section ??.

Two parameters determine how the LOF method performs; the  $k$  number of points considered as the nearest neighbors of each point and the threshold that regulates which points are outliers. Initial experiments showed that the lower the  $k$  value, the faster the performance of the method. Concerning the threshold, two different approaches have been applied and tested; a relative threshold and an absolute threshold. The difference between the two becomes clear by observing Algorithms 4.1 and 4.2. To apply the relative threshold (Algorithm 4.1), the computed LOF scores of all the points are sorted. Then, the relative threshold is multiplied by the total number of points to determine how many points are outliers. The number of outliers based on the relative threshold is notated as  $X$ . LOF scores significantly larger than 1 indicate outliers as mentioned in the theory (page 16). By sorting in decreasing order the high LOF scores appear first. Thus, the first  $X$  amount of LOF scores correspond to the outlier points.

---

**Algorithm 4.1:** LOF - Relative threshold

---

```

1 sorted ← Sort Decreasing(LOF scores (points))
2 N ← Amount Of Points
3 Amount Of Outliers ← N * Relative Threshold
4 X ← Amount Of Outliers
5 outliers ← sorted [From 0th to Xth]
6 inliers ← sorted [From Xth to Nth]
```

---



---

**Algorithm 4.2:** LOF - Absolute Threshold

---

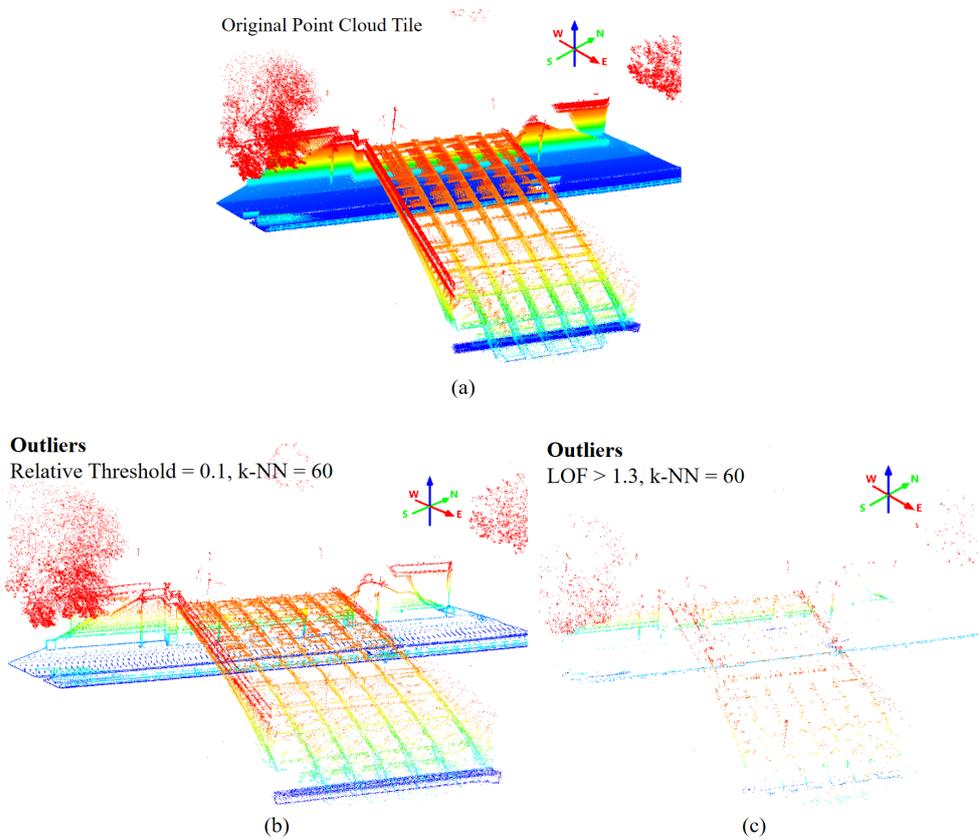
```

1 for LOF score (pointi) ∈ LOFscores(points) do
2   if LOF score (pointi) < Absolute Threshold then
3     | point ← inlier
4   else
5     | point ← outlier
6   end
7 end
```

---

The execution of some initial experiments proves that the absolute threshold is a better option. Figure 4.3a shows a point cloud tile. The outliers are visualized to have a clear impression of the rejected points. Figure 4.3b illustrates the outliers when a relative threshold equal to 0.1 is used. Figure 4.3c illustrates the outliers when an absolute threshold is used. Although the two thresholds cannot be compared numerically, it can be seen that with the relative threshold not only noise is rejected, but also the edges of the objects and the edges of the tile. This happens because points at the boundaries do not have neighbours in every side, thus they are considered isolated at some extension and as a result, their LOF score is a bit higher than 1. Also, their LOF score is lower than 1.3 as the points on the edges are not rejected with absolute threshold equal to 1.3. Moreover, the probability to remove same points with the a relative threshold from point cloud tiles

representing the same area but retrieved from different viewpoints is lower. This is because the amount of points in overlapping point clouds may differ significantly, and the rejection of outliers with the relative threshold depends on the total number of points in a point cloud. In contrast, in with absolute threshold the probability to remove same points from overlapping tile is higher as the rejection depend on the **LOF** scores.



**Figure 4.3:** a) A point cloud tile with 2380094 number of points. b) The **LOF** scores of the points in Figure (a) are computed. A relative threshold equal to 0.1 is applied and thus the number of outliers is 238009. c) The **LOF** scores of the points in Figure (a) are computed. An absolute threshold is applied. The outliers, which are illustrated, are the points of which the **LOF** score is higher than 1.3.

To sum up, the absolute threshold is preferred over the relative threshold because then the rejection of the outliers is based on the examined property; the local density of the points.

### 4.3.2 Normal vectors computation

**REASONING** After the outlier rejection, the normals vectors are computed on the points of each tile as they are needed for the generation of images from the point clouds (Section 4.4). In order to compute the normal vectors of the surfaces at each point, the PCA method is applied which is explained in Section 2.4.3. PCA constitutes a simple method for calculating normal vectors and therefore it is used.

**METHOD** The Algorithm 4.3 indicates how the normal vectors are computed with PCA. Firstly, the neighboring points of each point in a tile must be computed. These have been detected in the previous step for the rejection of the outliers, thus the same neighboring points are used for the computation of the normal vectors. The covariance matrix of each neighbourhood is computed by as explained in the theory and the eigen values and eigenvectors of the covariance matrix are computed. The smaller eigenvalue is found and its corresponding eigenvector is used as the normal vector that can be fitted by the neighbourhood of points. This vector is also the normal vector of the 3D point. However, the result does not provide consistent normal vectors. The normal vectors of one surface are not pointing in the same direction as they should. For example, some of the normal vectors of points that represent a wall point inside the building and some outside the building.

---

#### Algorithm 4.3: Normal vectors on 3D points

---

```

Input: 3D points
Output: list of normal vectors
1 for  $point_i \in 3Dpoints$  do
2   | find neighbors
3   | compute their covariance matrix
4   |  $minValue \leftarrow$  minimum eigenvalue(covariance matrix)
5   |  $PlaneNormalVector \leftarrow$  the eigenvector of the  $minValue$ 
6   |  $PointNormalVector \leftarrow PlaneNormalVector$ 
7   |  $createdVector \leftarrow$  vector from the trajectory point to the point cloud point
8   | if  $dotproduct(PointNormalVector, createdVector) > 0$  then
9   |   | add the normal vector to the list of normal vectors
10  | else
11  |   |  $normal\ vector \leftarrow -(normal\ vector)$ 
12  |   | add the normal vector to the list of normal vectors
13  | end
14 end

```

---

It is desired that the vectors are pointing at the same direction else the values of vectors of a plane will be different. Thus, in such case the normal vectors will be of no use. The trajectory points of the moving vehicle are used to orientate the normal vectors towards the vehicle's direction at the moment of the recording. For each point in the tiles there is a corresponding point recorded in the trajectory points. To orientate the normal vector of a point its corresponding viewpoint is used. A vector is formed from the examined point in the point cloud and its view point. Then the orientation of that vector is compared with the orientation of the computed normal vector by computing their dot product. If the output of the dot product is positive then the computed normal vector points at the direction of the trajectory. If the output of the dot product is negative then the computed normal vector points has incorrect orientation. Therefore the sign of the vector value is changed; from positive to negative or from negative to positive. In such a way, the normal vectors are pointing opposite to their initial direction.

### 4.3.3 Pairing overlapping point clouds

The following step is to pair overlapping point cloud tiles as the purpose of this project is to perform pairwise registration. Firstly it is required to know which point clouds overlap. Despite the fact that the positioning of the 3D data could be degraded due to the lack of GNSS reception, it is still possible to detect the overlapping point cloud sets from their coordinates by assuming a buffer zone. In the data-sets used it was already determined which point cloud tiles overlap. Therefore the only step left is the pairing of the point clouds. To do so, all the possible combinations of overlapping point clouds are made. The pair of point cloud 1 and point cloud 2 is the same as the pair of the point cloud 2 and the point cloud 1. This redundancy is not wanted. The number of possible combinations is given by the equation 4.1. For example, if there are 10 point cloud tiles that capture the same scene, but retrieved at different times from different viewpoints, then 45 different pairs of point cloud tiles are made.

$$\text{combinations} = n! / (r! * (n - r)!) \quad (4.1)$$

where  $n$ : the number of overlapping point cloud tiles  
 $r$ : number of point cloud tiles to combine. It equals to 2,  
 as point cloud pairs must be constructed.

## 4.4 CONVERSION FROM 3D TO 2D

This section explains how 2D imagery is acquired from 3D point cloud tiles with the proposed method, in a way to facilitate the 2D matching process. Initially it is discussed how the three dimensions of the data are reduced in two (Subsection 4.4.1). A thorough description of the illustrated attributes on the created imagery follows in Subsections 4.4.2 - 4.4.7.

### 4.4.1 Reduction of dimensions

**REASONING** For the reduction of the data dimensions two methods are found in the literature. The first method suggests the projection of the 3D data from a perspective point of view [Liang et al., 2017]. The second method suggests the orthographic top view projection of the 3D data to create a 2D XY-plane, but also the orthographic projection in 1D to represent only the Z coordinates [van Someren, 2016]. Two visual comparisons of a perspective and an orthographic projection are shown in Figures 4.4 and 4.5.

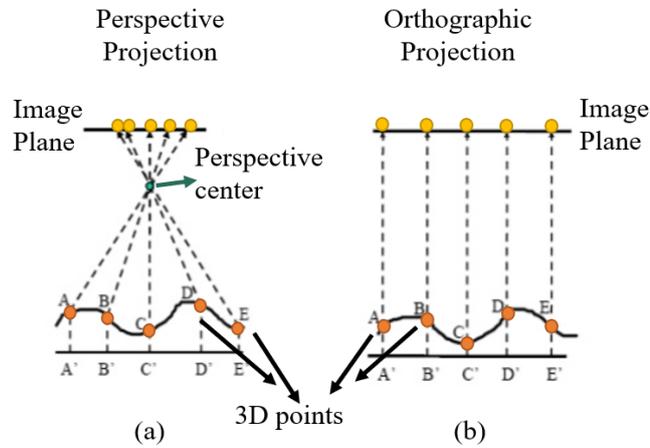


Figure 4.4: a) How to project a 3D surface from a perspective point of view. b) How to project a 3D surface from an orthogonal point of view.

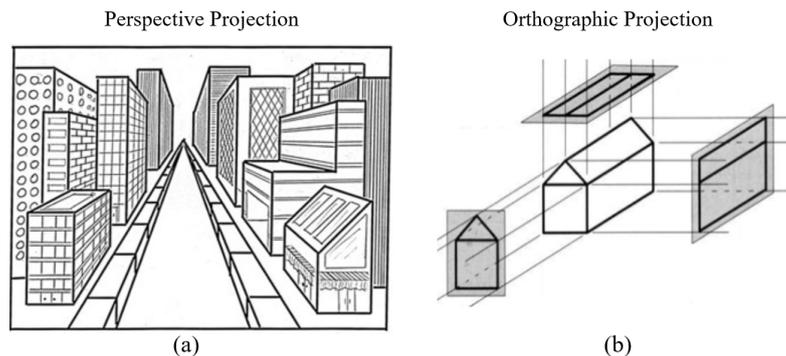


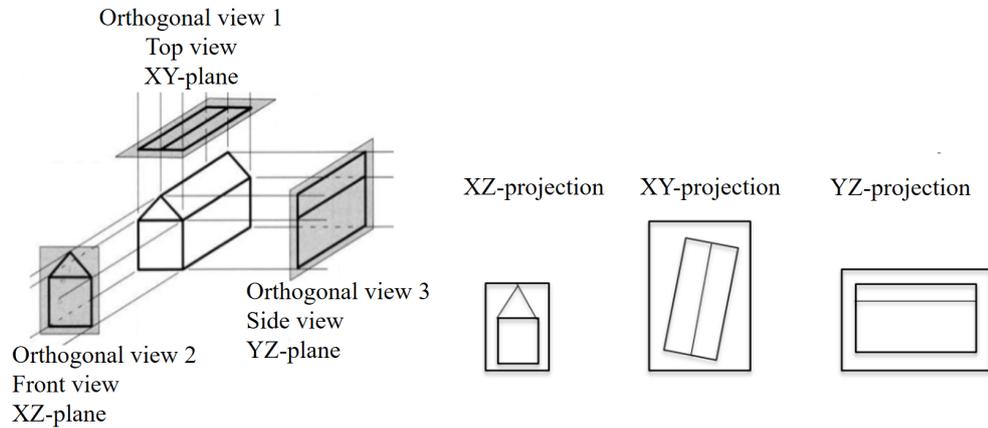
Figure 4.5: a) A perspective image of 3D objects. b) Three orthogonal images of a 3D object.

As it can be seen in both Figures, the parallel lines in a perspective projection appear to converge on a vanishing point, while the parallel lines in an orthographic projection never converge. The advantages and disadvantages of the perspective and orthogonal images are discussed in Table 4.1. Due to the disadvantages that the perspective images have, they are not used in this project.

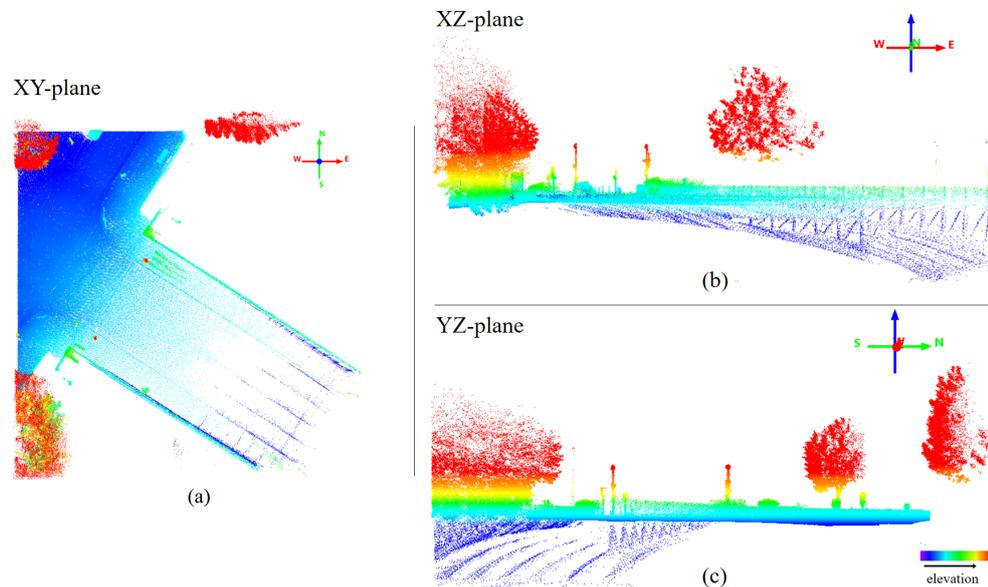
Perspective projections	Orthogonal projections
+ They can provide more information about the depth of the objects.	+ The depth may be incorporated in the pixels values.
- However, the depth information is implicit and not easily extracted.	
	<p>In general, the computational complexity of both images is the same, because the number of resulted pixels, which is the biggest performance factor, would be the same.</p> <p>+ However, it is possible to have less computational complexity when creating orthogonal images than when creating perspective images. This could be achieved by choosing a very simple setup. If a point cloud is just observed from a top view, then the only step to construct the orthogonal image would be to use the X and Y coordinates of the 3D points. No calculations would be needed to create the orthogonal image.</p> <p>Contrarily, to create a perspective image from a top view it is necessary to compute the 2D coordinates by using the collinearity equation, the central mathematical expression in photogrammetry.</p>
<p>The selection of the center of projection is critical because it will determine the overlap between the two point clouds.</p> <p>- If the center of projection is placed outside the point cloud tile area many unnecessary computations of 3D coordinates to 2D must be conducted. This will happen because many 3D points will end up having the same 2D coordinates. The same problem will occur if the center of projection is placed inside the point cloud, but in a lower degree.</p> <p>- If the center of projection is placed inside the point cloud tile area, for example on a trajectory point of one of the trajectories of the two point clouds, many images should be created to represent one tile and assure that there is overlap between the created images. Instead of creating several images, a panoramic (complete perspective) image could be used.</p>	

**Table 4.1:** A comparison between the perspective and orthographic projections concerning the image registration.

For the conversion of the 3D data to 2D a method that originates from the method used by van Someren [2016] is applied. The difference is that instead of one orthogonal 2D projection and one orthogonal 1D projection, each point cloud tile is projected on three different 2D planes resembling three orthogonal views. Each point cloud tile is viewed once from the top, from the front and from the side. Consequently, the point cloud tiles are projected in XY-planes, in XZ-planes and in YZ-planes. An overview of the concept of the projections is illustrated in Figure 4.6. Additionally, Figure 4.7 shows a point clouds tile which is viewed from the three planes used for the projections.



**Figure 4.6:** The figure illustrates how the 3D data are projected in 2D in three different planes. One plane is formed when the 3D data is viewed perpendicularly from the top, one plane is formed when the data is viewed perpendicularly from the front and one plane is formed when the data is viewed perpendicularly from the side.



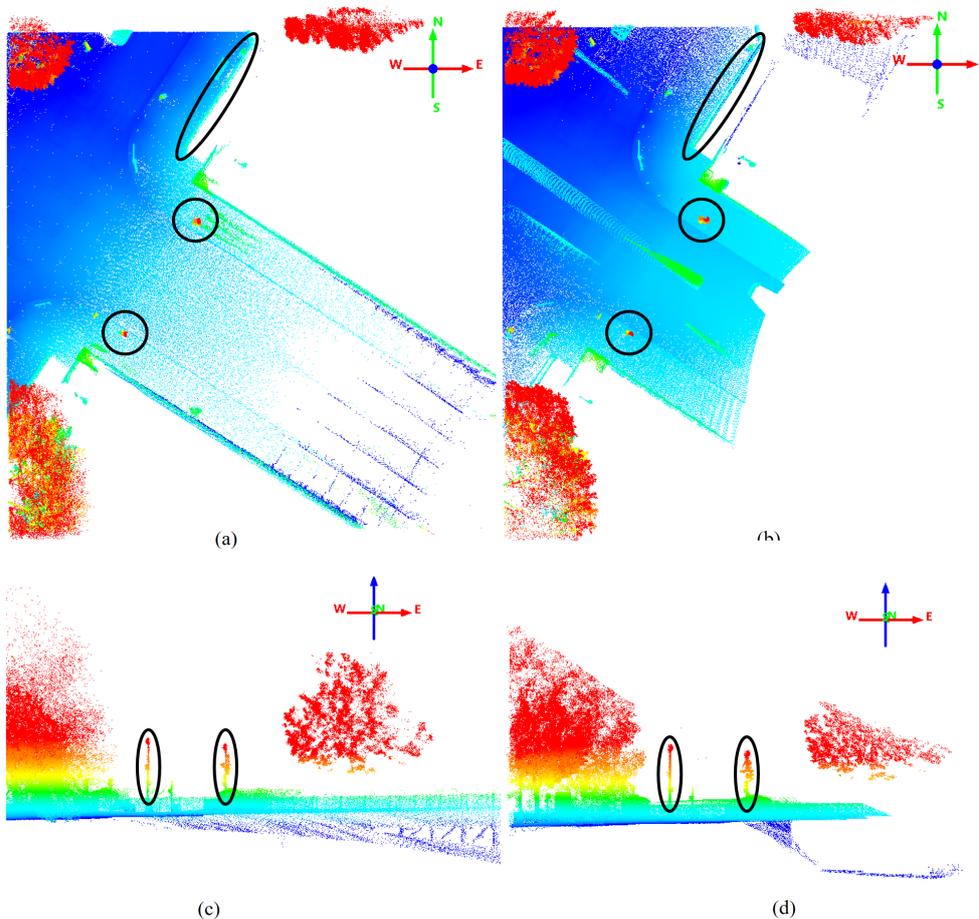
**Figure 4.7:** a) A point cloud tile viewed from the XY-plane. b) A point cloud tile viewed from the XZ-plane. c) A point cloud tile viewed from the YZ-plane.

By creating three 2D projections the method is benefited in three ways.

1. The translation parameter for X, Y and Z can be retrieved twice. By matching the XY-projections of two overlapping point clouds the translation in X and the translation in Y will be estimated. By matching the XZ-projections of two

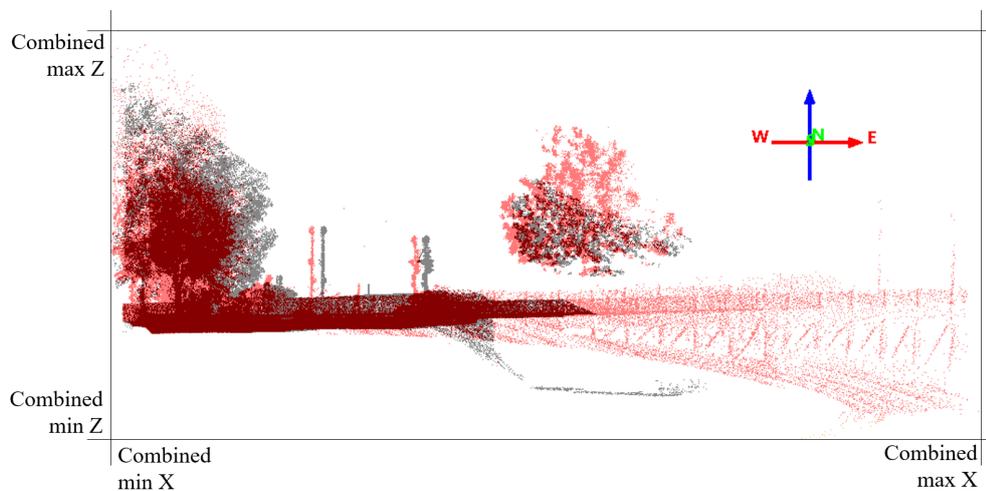
overlapping point clouds the translation in X and the translation in Z will be estimated. Similarly for the YZ-projections the translation in Y and the translation in Z will be estimated. Therefore in case one of the pair projections fails to estimate the correct parameters then there is still chance to get the correct result from another projection.

2. By generating three projections the possibility to have more common visual information in the generated 2D data is higher than by having one 2D projection. As mentioned in page 62, image registration is the process of matching imagery with common visual information. Therefore, in order to match an *image pair* the corresponding objects must be visible in the created projections. It is possible that common objects cannot be seen from one plane or that common objects can be seen better from one than from another plane (Figure 4.8). If 1D projections of the Z coordinate was used as in [van Someren, 2016] and [Christodoulou, 2017], then this advantage would not exist.



**Figure 4.8:** a) and b) Two overlapping point clouds viewed from the XY-plane. The black circles and the ellipsis indicate common visual information that could facilitate the registration. c) and d) The same overlapping point clouds as in a) and b) are viewed from the XZ-plane. The poles in the black ellipses indicate common information. These correspond to the objects circled in a) and b). In a) and b) the poles are shown like dots but in c) and d) they easily stand out. Therefore, the XZ-planes may produce better results due that. However the structure included in the black ellipsis in a) and b) cannot be seen by c) and d). This may facilitate the matching of the XY-planes.

**METHOD** The 2D projections are created per [point-cloud pair](#). For every point cloud pair the combined minimum and maximum X, Y, and Z coordinate is computed. These are used as the spatial boundaries for two 2D projections generated from the two corresponding point cloud tiles in a pair. Therefore, the two images resulted from one point cloud pair will have the same size. By using the same boundaries to create two images that correspond to two overlapping point cloud tiles, and specifically by using the same origin point, it is ensured that the points' coordinates are transported to the imagery. Thus, it becomes feasible to spot and compute the positioning offsets in the created image pairs. According to the plane of projection, the corresponding combined boundaries of the two point cloud tiles are utilized. For example, for the creation of the XZ-plane the combined boundaries of the X coordinates and the combined boundaries of the Z coordinates are used (Figure 4.9).



**Figure 4.9:** The red points are from one point cloud tile and the gray points are from another overlapping point cloud tile. The Figure illustrates the combined boundaries of the point cloud pair for the XZ-plane. Because the combined boundaries are used, the positioning offsets are distinct.

The combined boundaries of the 2D coordinates and a user specified grid cell size, are used to construct the 2D grid of each image. For the example shown in Figure 4.9, a XZ grid must be constructed. Let's assume that the combined minimum X boundary is 10m, the combined maximum X boundary is 60m and the grid cell size (pixel size) is 1m. Then it means that 50 columns  $((60 - 10)/1)$  be constructed at every 1m. The same procedure follows for the Z axis, for its combined minimum and maximum, in order to construct the rows of the grid. As a result, the 2D grid is constructed. By using the created grid for each point cloud in a pair, the 2D-coordinates of the two point clouds are used to spatially bin the points into the 2D grid cells (Figure 4.10, Figure 4.11) and create projections for each point cloud.

The concept of the reduction of the 3D points dimensions in order to bin the points in 2D grid cells based on the projection plane is summarized in Algorithm 4.4. In the pseudo code the coordinates in the two dimensions of a projected plane are noted as CoordinatesA and CoordinatesB. For example for the XZ-plane the CoordinatesA refer to the X coordinates of the points and the CoordinatesB refer to the Z coordinates of the points.

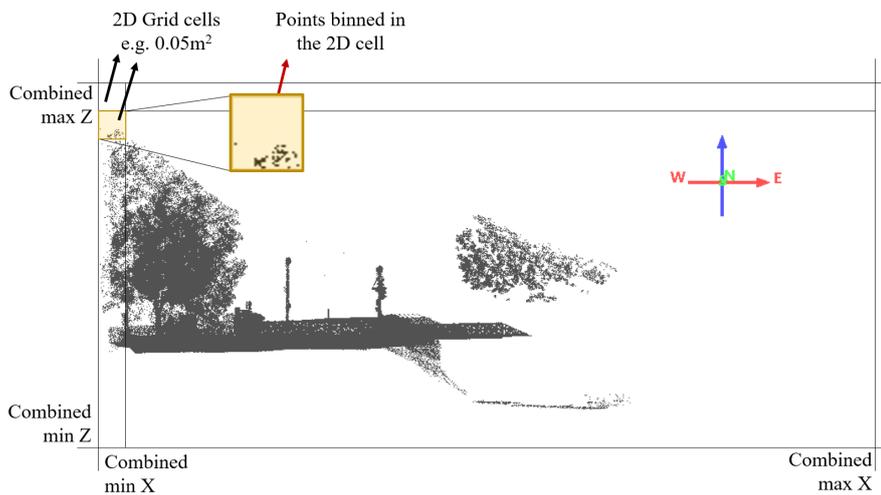


Figure 4.10: Binning the 3D points of one of the point clouds shown in Figure 4.9 in a 2D grid based on its X and Z coordinates.

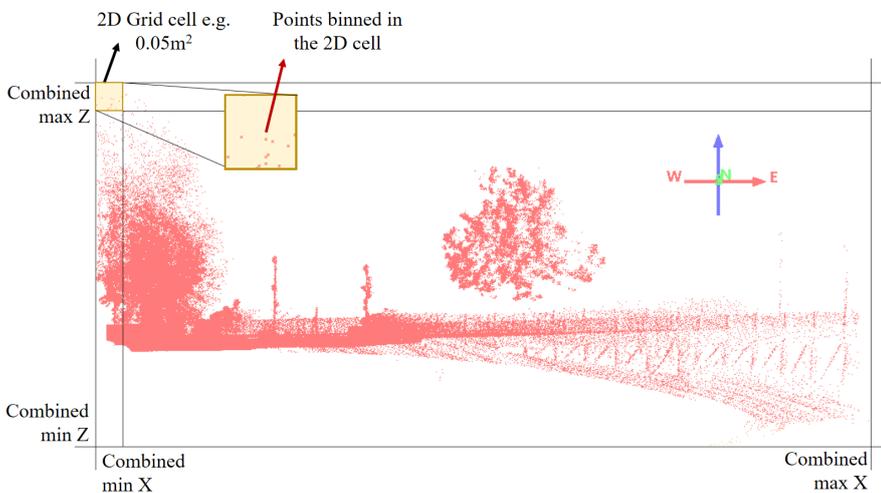


Figure 4.11: Binning the 3D points of the second point cloud shown in Figure 4.9 in a 2D grid based on its X and Z coordinates.

---

**Algorithm 4.4:** Binning the 3D points in a 2D grid

---

**Input:** Point Cloud 1, Point Cloud 2, Grid Cell Size, projection plane

**Output:** Binned Coordinates In 2D Grids

- 1 Use Coordinates A and Coordinates B based on the projection plane
  - 2 Find Min Coordinate A based on Point Cloud 1 & Point Cloud 2
  - 3 Find Min Coordinate B based on Point Cloud 1 & Point Cloud 2
  - 4 Find Max Coordinate A based on Point Cloud 1 & Point Cloud 2
  - 5 Find Max Coordinate B based on Point Cloud 1 & Point Cloud 2
  - 6 Construct the 2D grid for each point cloud tile based on the combined coordinates and the Grid Cell Size
  - 7 Bin the points of Point Cloud 1 in the 2D grid
  - 8 Bin the points of Point Cloud 2 in the 2D grid
-

**ASSIGNING VALUES TO THE 2D GRID CELLS** Every time that the 3D points are binned in the 2D grid cells, one pixel value is simultaneously computed. Attributes of the points are used as the illustrated information on the 2D-projections. In other words attributes of the points provide the pixel values on the grid cells of the generated projections. The attributes used aim to describe the 3D information in 2D. Since many points may be binned in one grid cell, the attributes of the points are compressed to one value by using statistics such as the mean and the max.

In this project, the three projections that are used to reduce the dimensions of the data are generated multiple times with different attributes of the 3D points. The projections are created with pixel values representing the density of the points within the grid cells, the intensity, the depth, the gradient of the intensity, the gradient of the depth and the calculated normal vectors of the points. In other words, a set of images is produced from every point cloud tile. This decision was taken in order to increase the reliability of the results. More information is found in Section 4.7.

As mentioned on pages 48 - 49, the binning of the points in 2D grid cells is performed per [point-cloud pair](#) so that the translations between the [image pairs](#) can be ultimately detected. Likewise, the assignment of points' attributes into the grid cells of the 2D created projection is performed per point cloud pair. This gives the capability to use properties of both projections in order to assist the next step, the image registration. Details are found in the following subsections when the methods used for assigning attributes to the pixels are described.

#### 4.4.2 Density images

The density of the points is used as one attribute to fill the grid cells of the generated projections on the three planes. Each grid cell of a density image simply illustrates the total amount of points that fall in the 2D cell.

**REASONING** The density is considered an important characteristic because long and thick features, such as walls or the ground, will be represented with high amounts of points. Thus these features can be highly distinguishable in image pairs.

**METHOD** In order to create a density image a histogram is computed from the 2D grid cell of the generated projections and the points that are binned to each cell. Therefore a single value is provided for each pixel. This value is amount of the points that fell in the cell of the grid. As only one value is used to color the pixels, only one color channel is used and thus, the density images are in grayscale. The pixels of grayscale images illustrate only shades of gray. Black pixels represent the lowest possible intensity while white pixels represent the highest possible intensity.

The downside of a density image with respect to an image matching process is the fact that corresponding objects in two overlapping images may be described by more or fewer points. The number of recorded points depend on the recording vehicle's position towards the object. The closer the moving platform to the object, the more the points that will be recorded, while the further the moving platform to the object, the fewer the points. Examples of density projections are illustrated in Figure 4.12.

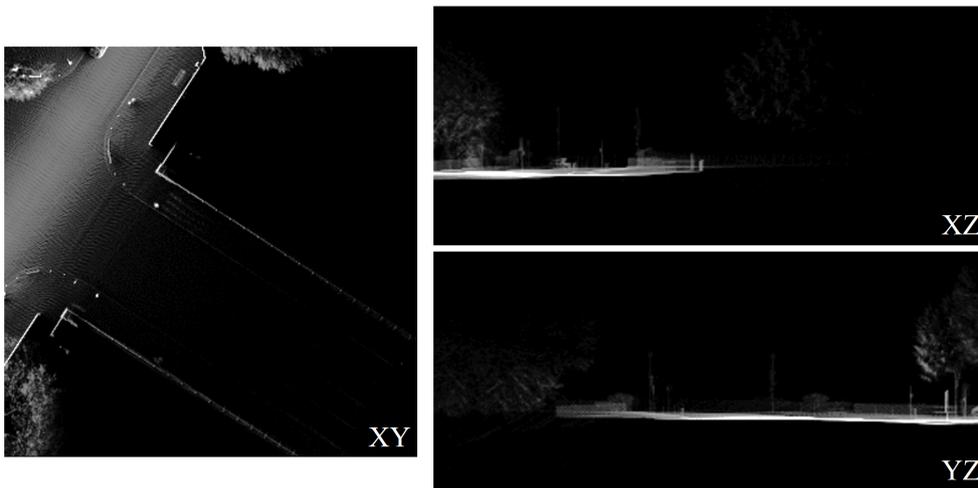


Figure 4.12: The three images produced from the point cloud tile of Figure 4.7 illustrating density values. The images have grid cell size 0.15m.

#### EXPLANATION OF THE IMAGES

- By observing the Figure 4.12 very bright and very dark pixel can be seen. The strongly bright pixels indicate that in the specific grid cells the number of recording points was high. The strongly dark pixels indicate that in the specific grid cells the number of recording points was low.
- In the XZ and YZ projections the most distinct feature is the ground. To create for example the XZ projection, the tile is sliced in the X (horizontally) and Z (vertically) direction. All the points of the 3D tile are binned in the 2D cells based on their X and Z coordinates. The 2D cells of the horizontal slice that the ground points are binned are filled with many points and thus, they are very bright.
- Even if the density values of an [image pair](#) are normalized up to the same value, it is possible that pixels of corresponding objects will have very different values. This is clear by observing the Figure 4.13.

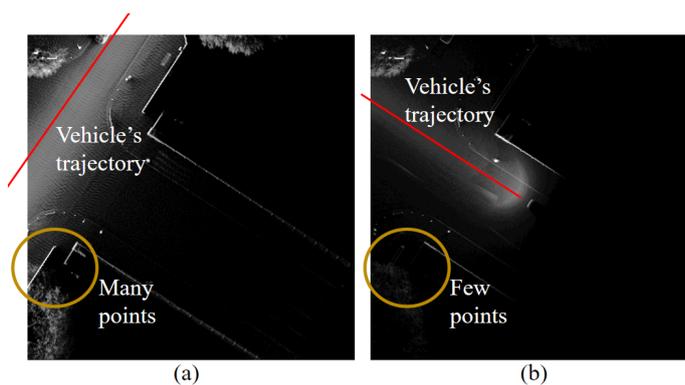


Figure 4.13: The two images illustrate the density images created from two overlapping point clouds. As it can be seen, the recording vehicle scanned the same area by following different trajectories. Consequently, different number of points collected for some identical objects. For example the objects which are circled with yellow in Figure (a) are very bright indicating many points binned in the 2D cells. The objects which are circled with yellow in Figure (b) are very dark indicating few points binned in the 2D cells.

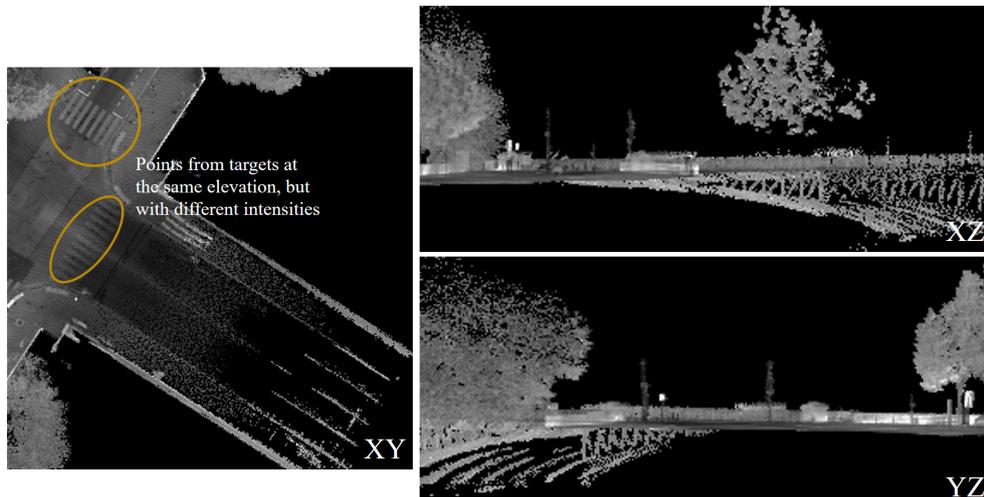
#### 4.4.3 Intensity images

The three projections of each point cloud tile are also created with the points' intensity values. The laser beams of the scanner are emitted and interact with the surrounding surfaces. Then, backscatter is generated and the signals are received back by the scanner. The intensity of a point refers to the strength of the corresponding received laser signal [Höfle and Pfeifer, 2007].

**REASONING** Godin et al. [1994] used the intensity of the points in order to assist an ICP point cloud registration process. Also, Pfeifer et al. [2007] state that intensity measurements are mostly used for the visual analysis of point clouds but they should be used for other process like registration. But how can the intensity facilitate a point cloud registration process? The strength of the backscattered signal is depended on the range between the scanning system and the object that scatters back the emitted beam, the angle of incidence between them and the object's reflectivity Pfeifer et al. [2007]. The object's reflectivity can be very useful for registration. Objects that lay exactly on same 2D plane but have high reflectivity become distinct in intensity images.

**METHOD** The intensity values are stored along with the 3D point. Thus, when the 3D points are retrieved, the intensities are also read to construct the intensity images. To produce one intensity value per grid cell, a statistical measure is used which considers the intensities of all the points that are binned in a cell. The results produced by using different statistics are influenced by the selected grid cell size. For example, if a large cell size is used then the mean of the intensity values in a cell will be very incorrect. It will be the mean of very different normal vector values. In such case, it would probably be better to use the mode of the collected intensity values. For very small grid cells it is assumed that the influence of the statistic on the registration results will not be great. Different statistical measures are applied and compared in Chapter 5 (Implementation & Experiments).

Examples of intensity projections are illustrated in Figure 4.14.



**Figure 4.14:** The three images produced from the point cloud tile of Figure 4.7 illustrating intensity values. The images have grid cell size 0.15m and each pixel value is the result of the mean of the binned points' intensities.

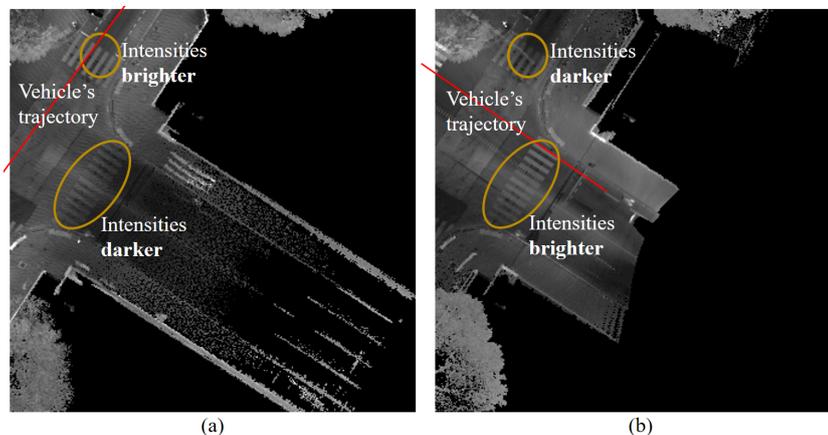
#### EXPLANATION OF THE IMAGES

- The brighter the pixel, the higher the intensity value. High intensity values correspond to objects with high reflectivity. These objects have brighter colours.
- The darker the pixel, the lower the intensity value. Low intensity values correspond to objects with low reflectivity. These objects have dark colours.
- For example, as it can be observed in the circled areas of the Figure 4.14, the painted lines on the roads which usually have white or yellow color are represented with brighter pixels. In contrast, the rest of the road surface is represented by darker pixels.

#### 4.4.4 Gradient of intensity images

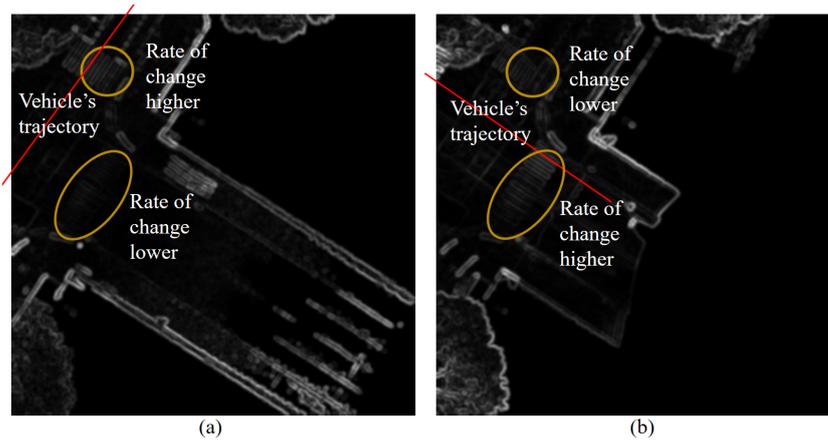
Apart from the intensity images also images that illustrate the gradient of the intensity are generated.

**REASONING** Corresponding points in two overlapping point cloud tiles may have different intensities. This happens because the intensity values depend among others on the distance between the scanned object and the position of the scanning system at the time of the recording. The Figure 4.15 illustrates the problem. Images that illustrate the gradient of intensity are introduced to support the multi-registration method. As explained in the theory in Subsection 2.5.1, the information expressed by the gradient is the rate of change of the values in one image, in the horizontal and vertical direction. As it was clearly shown in Figure 2.10d of page 2.5.1, when the gradient of an image is computed the edges of the objects are clearly distinguished. By computing the gradient of intensity, the problem of the intensity images is not solved as explained with Figure 4.16.



**Figure 4.15:** The two Figures illustrate the intensity XY-projection created from two overlapping point cloud tiles. The pixels in the yellow circles show corresponding areas in the two images where the intensity values are different. This depends on the distance between the objects and the recording vehicle. When the points are scanned from a short distance, the pixels in the intensity images are brighter. When the points are scanned from a long distance, the pixels in the intensity images are darker.

The added value is the provision of a new types of images which is assumed that can help the registration process. This assumption is made because the rate of change is a property which is not visualized in the intensity images. It is of high importance because it indicates the boundaries of the objects, which it is believed they can determine the registration results.



**Figure 4.16:** The two Figures illustrate the gradient of intensity XY-projections generated from two overlapping point cloud tiles. As the gradient expresses the rate of change in the values, the gradient has very high values when the rate of change is high. The areas in the circles are observed and compared with the areas in the circles of the intensity images (Figure 4.15). When the intensities of the stripes on the roads are high, then the rate of change in the gradient of intensity is high. When the intensities of the stripes on the roads are low, then the rate of change in the gradient of intensity is low.

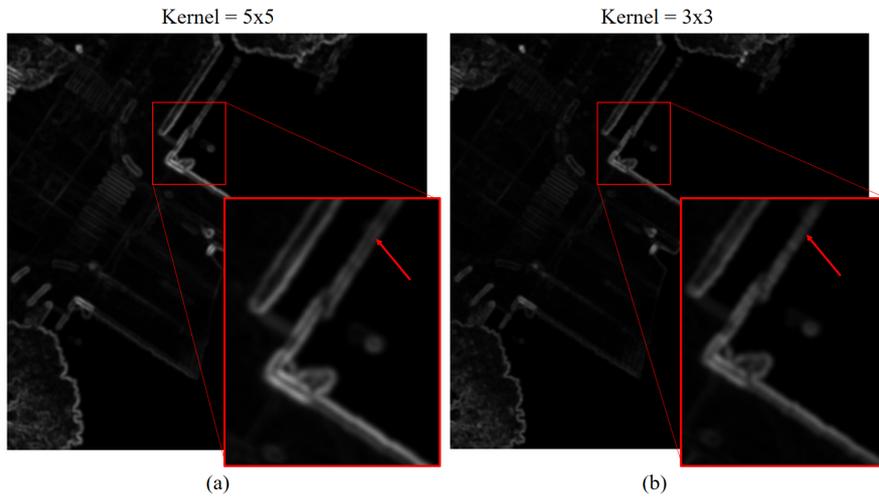
**METHOD** As mentioned in the theory the computation of the gradient of an image requires the computation of the first derivative of the image. The Sobel operator is a very common operator suitable for approximating differentiation. The approximation is performed by fitting discrete values on a polynomial. The Sobel operator is applied on the intensity images created on the previous step to compute the gradient of the intensity. Relevant theory about the Sobel operator is found in Section 2.5.2.

The size of the kernel indicates the extent of the approximation since the number of the discrete values in the kernel are fitted on a polynomial. If a large kernel is used then the kernel is computing the fit over a larger area of pixels. Therefore, a large kernel will have higher possibility to compute a closer approximation to the differentiation. Consequently, a 5 X 5 pixels kernel is chosen. Prior to the explanation of the whole method, a Figure relevant to the choice for the kernel's size is discussed. Particularly, the method for the computation of the image gradient is applied twice, once with a 3X3 kernel and once with a 5 X 5 pixels kernel. As it can be seen in Figure 4.17 some edges of objects are better distinguished when the kernel's size is 5x5. This could positively assist the image registration process.

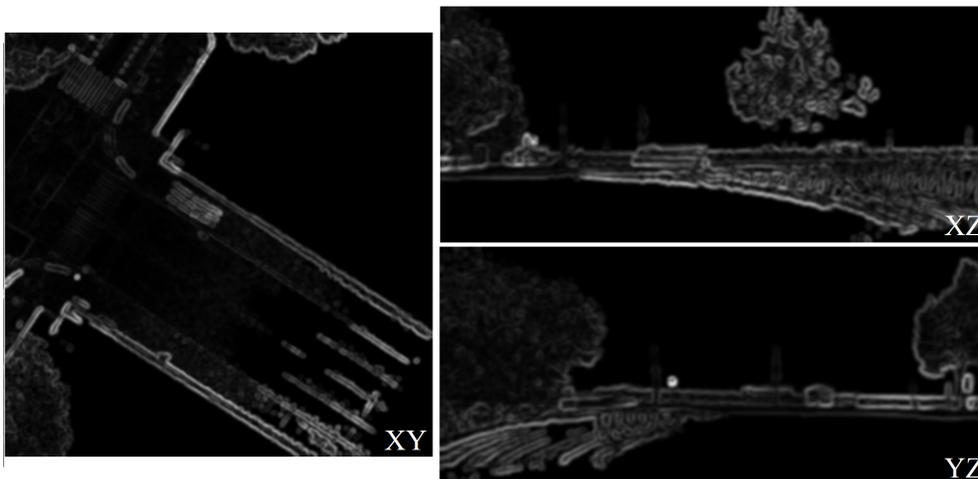
After the selection of the kernel size the gradient of each intensity image is computed once in the horizontal and once in the vertical direction by using the respective kernel. A software library is used to compute the gradient in X and in Y direction (Section 5.1.2). Next, the magnitude of the gradient is computed with the equation given in the theory. It was noticed that points that were very isolated, as expected, became more distinct when the gradient was computed. Such cases could confuse the registration, therefore a Gaussian smoothing filter was applied to the intensity gradient images. An example of the final intensity gradient images is illustrated in Figure 4.18.

#### EXPLANATION OF THE IMAGES

- The bright pixels indicate change in the intensity values. The dark pixels indicate no change in the intensity values.
- The brighter the pixel the bigger the change in the intensity values.



**Figure 4.17:** a) The gradient of the intensity applied on an XY-projection with kernel size 5 x 5 pixels. b) The gradient of the intensity applied on an XY-projection with kernel size 3 x 3 pixels. As it can be observed at the magnified part of the images and especially at the edges where the arrows are pointing, the result with a kernel 5 X 5 produces edges that are more clear.



**Figure 4.18:** The three images produced from the point cloud tile of Figure 4.7 illustrating gradient of intensity values.

- With the gradient of intensity the edges of the objects are distinguished. This is something that is not visible on the intensity images (Figure 4.14).

#### 4.4.5 Depth images

The attribute of depth of the 3D points is also used to assign values to the pixels of the XY, XZ and YZ projections. Depth images depict how far the objects of an image are with respect to a view point.

**REASONING** The depth of the 3D points is considered useful for the multi-registration process because identical objects placed at different depths can be distinguished. For example, if some identical buildings are placed next to each other but a few of them are placed at a shorter distance to the viewpoint than other buildings, then their depth values will be different. In such a way, the possibility that the image registration process matches identical buildings incorrectly would be lower.

**METHOD** The depth of each pixel in the created projections considers the coordinate of the 3rd dimension of the binned points. For example, the pixels of an XY-projection illustrate the Z-values of the points. The closer the Z from an imaginary XY-plane viewing the point cloud tile perpendicularly from the top, the less the depth of the points. Similarly, a YZ-projection illustrates the X-values of the points as their depth. The closer the X from an imaginary YZ-plane viewing the point cloud tile perpendicularly from the side, the less the depth of the points. Lastly, a XZ-projection illustrates the Y-values of the points as their depth. The closer the Y from an imaginary ZY-plane viewing the point cloud tile perpendicularly from the side, the less the depth of the points. In many LiDAR processing applications the depth value computed for 2D data is the distance value between the laser scanner and the point object. However, for a point cloud registration process such an action would not be ideal. Two overlapping point clouds tiles may have been retrieved from different viewpoints, thus corresponding points will not have the same depth.

To produce one depth value per pixel again different statistical measures are compared in Chapter 5. Examples of depth projections illustrating the maximum depth value calculated per pixel are given in Figure 4.19.

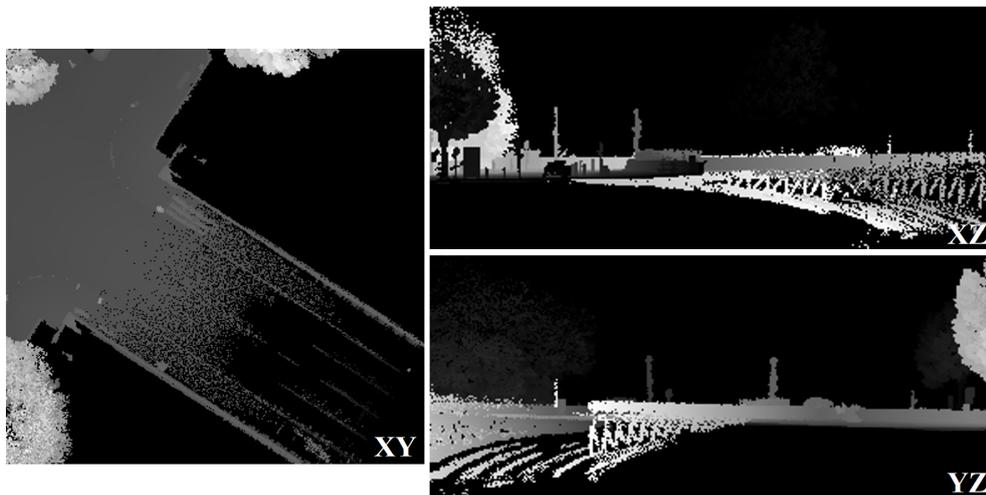


Figure 4.19: The three images produced from the point cloud tile of Figure 4.7 illustrating depth values. The images have grid cell size 0.15m and each pixel value is the result of the max of the binned points' depths.

#### EXPLANATION OF THE IMAGES

- The brighter the pixel, the less the depth of the binned points from an imaginary plane from which the points are observed.
- If the trees at the left part of the XZ-projection in Figure 4.19 are observed, one might think that the tree illustrated with black pixels is closer to the viewpoint than the tree illustrated with bright pixels. But as explained, bright pixels are closer to the viewpoint. The images show the maximum depth of the points collected in each cell. In the case of the XZ-projection the maximum Y coordinate lays the further away from the viewpoint of the XZ-plane. The tree illustrated with bright pixels overlaps the tree illustrated with black. But because the maximum depth is chosen, the tree at the back is also shown and not only the tree at the front. Where bright pixels exist, it is indicated that nothing else exist behind those pixels.

#### 4.4.6 Gradient of depth images

Apart from the depth images also images that illustrate the gradient of the depth are generated.

**REASONING** The drawback of the depth images is that the depth value of the same object in two overlapping point clouds may be different. That can happen because of the possible offset between the coordinates of two overlapping point clouds. For instance, if the depth image in a XY-projection is created and there is a positioning error in Z direction of 10cm that means that corresponding depth values in the two images will differ by 10 cm. Therefore, image registration techniques which are dependent on the common visual information of the matched images can possibly produce incorrect results. In order to make the multi-registration method profit from the depth information, images that illustrate the gradient of the depth are constructed. The gradient of the depth will make the depth discontinuities distinguished. Thus, even if the depth of two corresponding pixels in two overlapping images differs, the depth discontinuities will be the same. In such a way, the possibility to retrieve better matching estimates than with the depth images is considered higher.

**METHOD** The images that illustrate the gradient of depth are constructed exactly the same way as the images that illustrate the gradient of the intensity. An example of gradient of depth images is given in Figure 4.20.

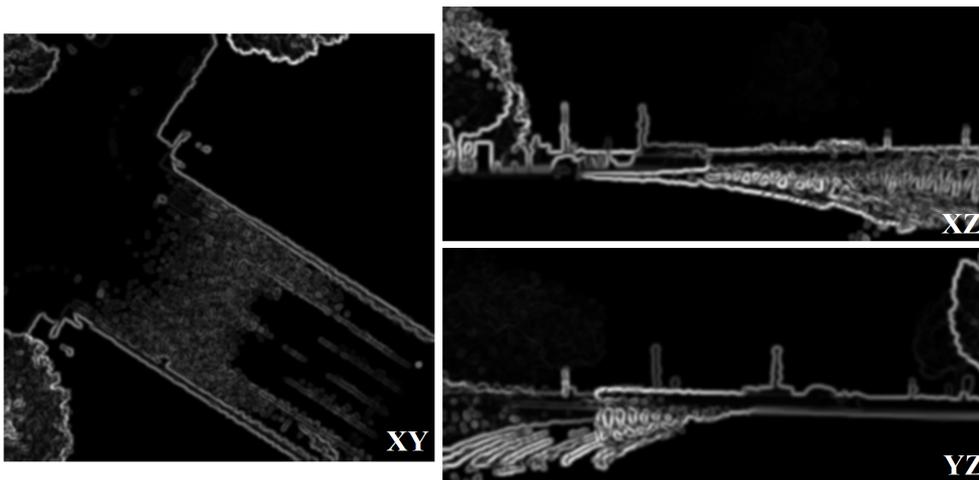


Figure 4.20: The three images produced from the point cloud tile of Figure 4.7 illustrating gradient of depth values. The images have grid cell size 0.15m.

#### EXPLANATION OF THE IMAGES

- The bright pixels indicate change in the depth values. The dark pixels indicate no change in the depth values.
- The brighter the pixel the bigger the change in the depth values.

#### 4.4.7 Images based on normal-vectors

The normal vectors computed on the points of each tile are also used to assign values to the pixels of the XY-, ZX- and ZY-projections.

**REASONING** The image registration approach could be assisted by using the normal vectors as the pixel values. As the normal vectors determine the surfaces'

orientation, the created images based on normal-vectors will have the same property. Instead of using the normal vector the bearing angles could be computed as suggested by Lin et al. [2017]. However, by comparing the images based on normal-vectors and the images based on the bearing angles, it is decided to use the normal-vectors. The advantages and the disadvantages between the two types of images concerning the image registration are summarized in the Table 4.2. The construction of the bearing angle images takes into account consecutive measurements of points to highlight the orientations of the surfaces on the created images. However, consecutive point measurements may not be exactly scanned on the edges of the objects. Consequently, the edges created with the bearing angle images will not be representative of the real edges of the objects.

Bearing angle images	Normal vector images
Capability to highlight the orientation of the surfaces at specific directions.	Capability to highlight the orientation of the surfaces at all directions.
Consecutive point measurements may not be exactly scanned on the edges of the objects. Consequently, the edges created with the bearing angle images will not be representative of the real edges of the objects.	

Table 4.2: A comparison between the images based on the bearing angles and the images based on the normal vectors.

**METHOD** The normal vectors computed for each point cloud in the pre-processing phase are used to create the images based on the normal vectors. Each normal vector value is consisted of three components; the value in X, Y and Z. To store the vector values, three color channels are used. Therefore the normal images are RGB-based. The values of the three components could be merged into one, thus the output would be grayscale images. In such a way the normal vector images would require less storage memory. However, this is not performed in this project, but it could be considered as a future enhancement of the method. The Figure 4.21 shows an example of the XY, XZ and YZ projections which illustrate the normal vectors of the points in the point cloud shown in Figure 4.7. To produce one normal vector value per grid cell, the mean of the normal vectors of all the points that are binned in each cell is calculated. The output is influenced by the selected grid cell size. For example, if a large cell size is used then the mean of the normal vectors in a cell will be very incorrect as it will be the mean of very different normal vector values. In such case, it would probably be better to use the mode of the collected normal vector values, but this is not currently implemented. For very small grid cells it is assumed that the influence of the statistic on the registration results will be insignificant.

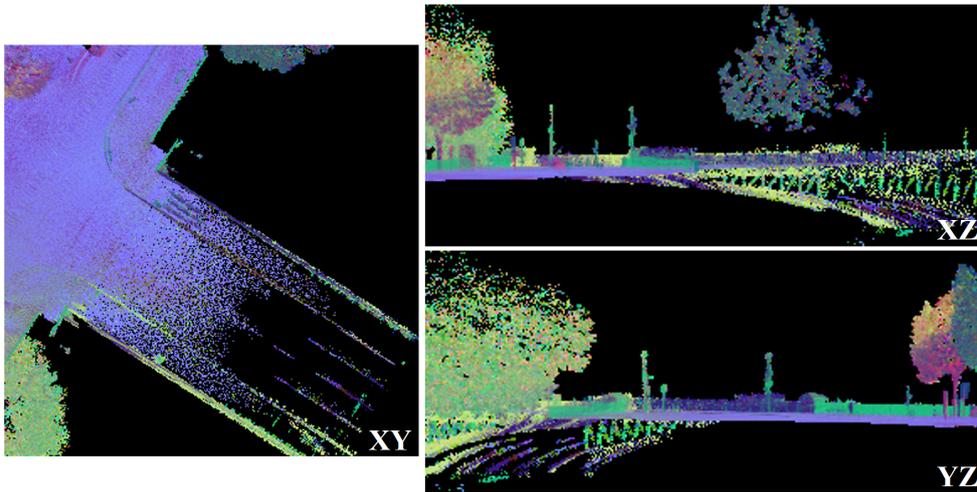


Figure 4.21: The three images produced from the point cloud set of Figure 4.7 illustrating normal vector values. The grid cell size equals to 0.15m.

#### EXPLANATION OF THE IMAGES

- The images show the perpendicular vectors of the planes that can be fitted on neighbourhoods of points. That means that the pixels which have the same color belong to surfaces of which the perpendicular vector is pointing at the same direction.
- For example it can be seen that the ground is filled with pixel of the same color (purple). That is because the normal vectors of the points at the ground are pointing towards the same direction, towards the sky.
- On the normal images the directional discontinuities can be spotted. For example green lines of pixel can be seen among the purple pixels that represent the ground (Figure 4.21, plane XY).
- Structures that are parallel and the recording vehicle passed in the area between can be spotted, as they are illustrated with different colours. For example, in the XY-plane there are parallel lines at the lower part of the image and approximately in the middle of the image. Those that are on the lower part have green colour and those that are in the middle have dark purple colour. This is an indication that the registration approach could not be confused by wrongly matching different structures that have identical form. The normal vectors of the parallel lines have same direction but different orientation because the vectors are pointing towards the trajectory of the recording vehicle.

#### 4.4.8 Images based on the components of the normal-vectors

Apart from the images based on the normal vector, the normal vectors are used to generate images that show the vector's value only in X, only in Y and only in Z direction.

**REASONING** The decision to create these images is taken because it is expected that in this way the surfaces' orientation could be strengthened. By using the vectors' values only in one direction it means the focus of the orientations lays on the selected direction. Further explanation follows.

**METHOD** To construct the images of the normal vector's components, each channel of the normal vector images is used separately to create the three different types of images. Similarly with the other types of images that have been mentioned, different statistical measures are used in order to produce one value per pixel. These are compared in Chapter 5. The Figures 4.22, 4.23 and 4.24 show an example of the relevant images for the point cloud shown in Figure 4.7. The Figure 4.22 shows the values of the normal vectors in X direction, the Figure 4.23 in Y direction and the Figure 4.24 in Z direction. A legend is added to the Figures to explain the symbols which are added in order to assist the explanation of the images.

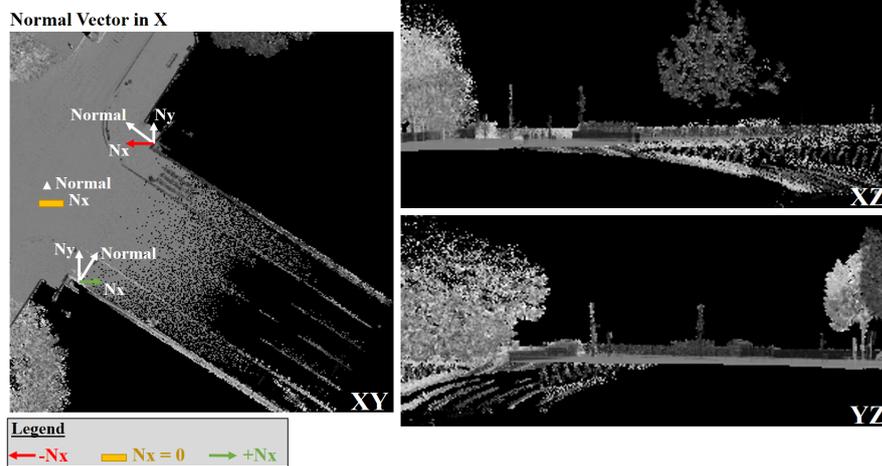


Figure 4.22: The three images produced from the point cloud set of Figure 4.7 illustrating the X values of the normal vectors.

#### EXPLANATION OF THE IMAGES

- If the selected component of the computed normal vector has high value then the pixel's value will be high. High pixel values are bright in grayscale images. If the selected component of the computed normal vector has low value then the pixel's value will be low. Low pixel values are dark in grayscale images. If the selected component of the computed normal vector has an average value then the pixel's value will be average. Average pixel values are gray in grayscale images. This is visualized on the XY-projections of the Figures 4.22, 4.23 and 4.24. Same explanation is valid for the other projections.
- For example in Figure 4.22 which shows the normal vector in X direction the normal vectors are drawn at three points. Then the normal are split in their components, the  $N_x$  which is the normal in the horizontal direction and the  $N_y$  which is the normal in the vertical direction. For now the  $N_z$  which is the normal in Z direction is skipped since the image illustrates the  $N_x$  values. The dark pixel values appear when the  $N_x$  has a low value, the bright values appear when the  $N_x$  has a high value and the gray values appear when the  $N_x$  has an average value. Same logi valids for the Figure 4.23.
- In Figure 4.24 which shows the normal vector in Z direction the normal vectors are drawn at two points. When the normal points towards the sky, the  $N_z$  has a high value thus the pixels are bright. When the  $N_z$  points  $-90^\circ$  or  $+90^\circ$  from the sky then the  $N_z$  has an average value and thus the pixels are gray. Dark pixel values do not appear as the normal vector never points towards the ground. This occurs because the normal vectors are always orientated towards the scanning system.

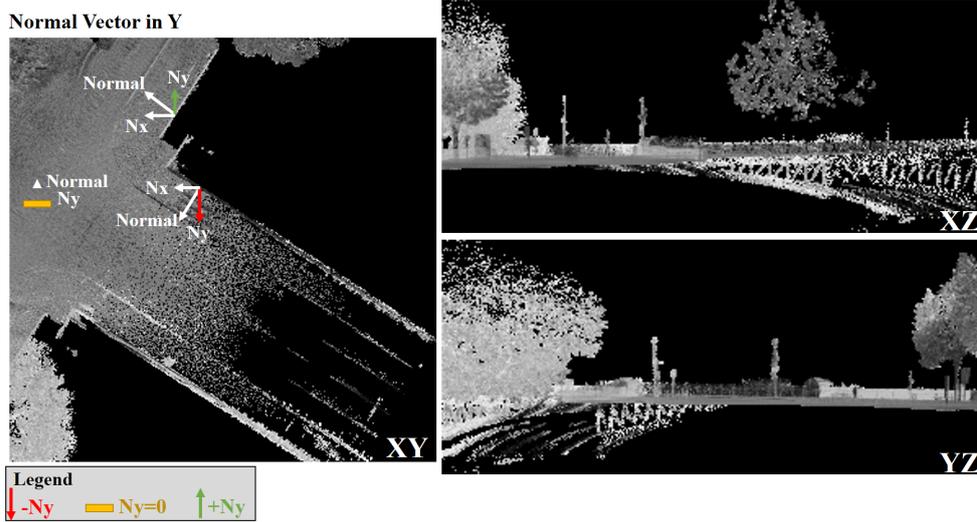


Figure 4.23: The three images produced from the point cloud set of Figure 4.7 illustrating the Y values of the normal vectors.

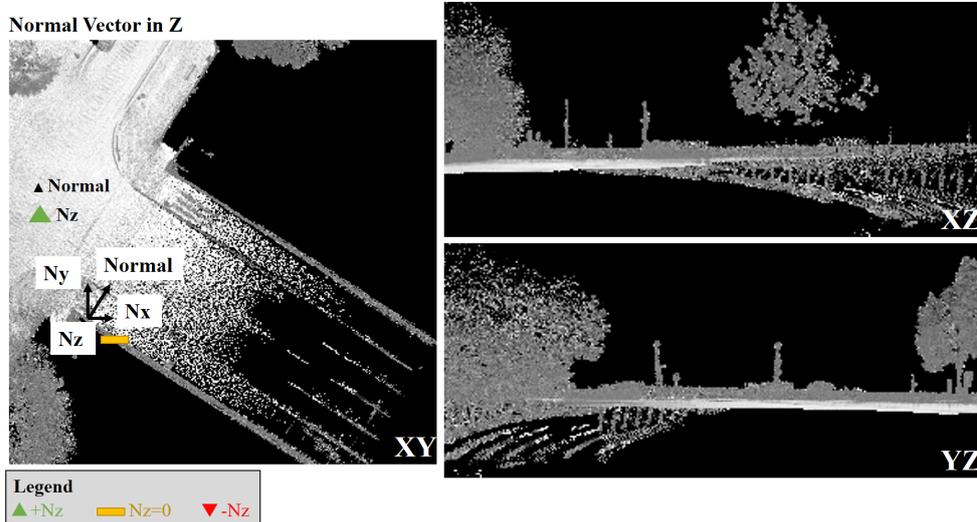
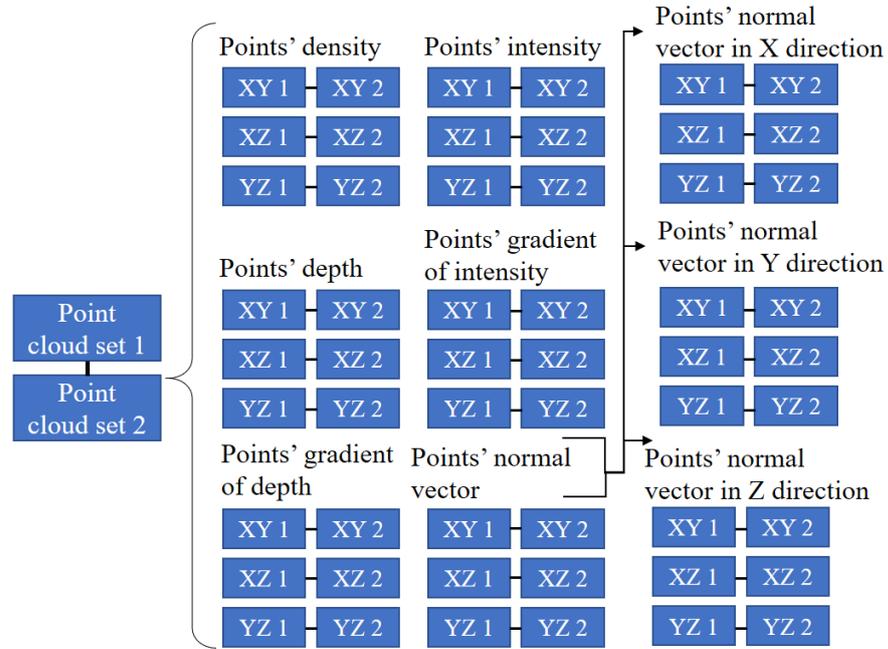


Figure 4.24: The three images produced from the point cloud set of Figure 4.7 illustrating the Z values of the normal vectors.

In this Section, the proposed method for the conversion of 3D to 2D data was presented. To summarize, nine types of images are used to represent each point cloud tile. An image type defines the points' attribute used to fill the 2D grid cells of the three projections. For each image type, a set of three projections is produced. For example, the XY, XZ and YZ projections that illustrate the density of the points define one image type. Overall, 27 image pairs are constructed for each point cloud pair (Figure 4.25).



**Figure 4.25:** The 27 produced image pairs from each overlapping point cloud pair. Each small blue rectangle indicates the plane of the projection. This numbering indicates the point cloud that is projected.

#### 4.5 PAIRWISE IMAGE REGISTRATION

For each pair of overlapping point cloud tiles, 54 images are created. The 54 images correspond to 27 images per point cloud tile. 27 image matchings are performed per point cloud pair, therefore the proposed method is often referred to a multi-registration method. A template matching technique is used to register [image pairs](#). Detailed theoretical information about template matching techniques is found in Section 2.5.5. Briefly, a template matching technique determines the location of a template image within a reference image [Sarvaiya et al., 2009]. The detection of the one image's location into the other is performed by comparing the pixel values of the two images. The best match lays at the pixel where the highest similarity value is computed.

The proposed method suggests having projections in XY, XZ and YZ planes for each image type. The registrations of the three projection pairs of the same type output twice each translation parameter. For instance, the registration of the density XY-projection of a point cloud tile  $t_1$  with the density XY-projection of a point cloud tile  $t_2$  outputs the X and Y translation parameters. The registration of the density XZ-projection of a point cloud tile  $t_1$  with the density XZ-projection of a point cloud tile  $t_2$  outputs the X and Z translation parameters. Similarly, the registration of the density YZ-projections outputs the Y and Z translation parameters.

**REASONING** As mentioned in Section 4.4 on page 50, the decision for creating multiple images per tile was taken in order to increase the confidence levels of the results. However, this design decision should not be the reason why the method becomes very complex and requires large computation time. There must be a balance. Template matching techniques consider simple statistical analyses of the brightness values of two images. Therefore, the template matching methods are suitable for the proposed multi-registration method. An alternative of template matching techniques could be the detection and matching of features in the images [Ruble et al., 2011]. As implied, such techniques require first the implementation of

a method to detect unique features and then their matching to compute the transformation parameters. In contrast, a template matching technique can be used to estimate the position of the one image into the other only by comparing statistically the pixel values of the two images.

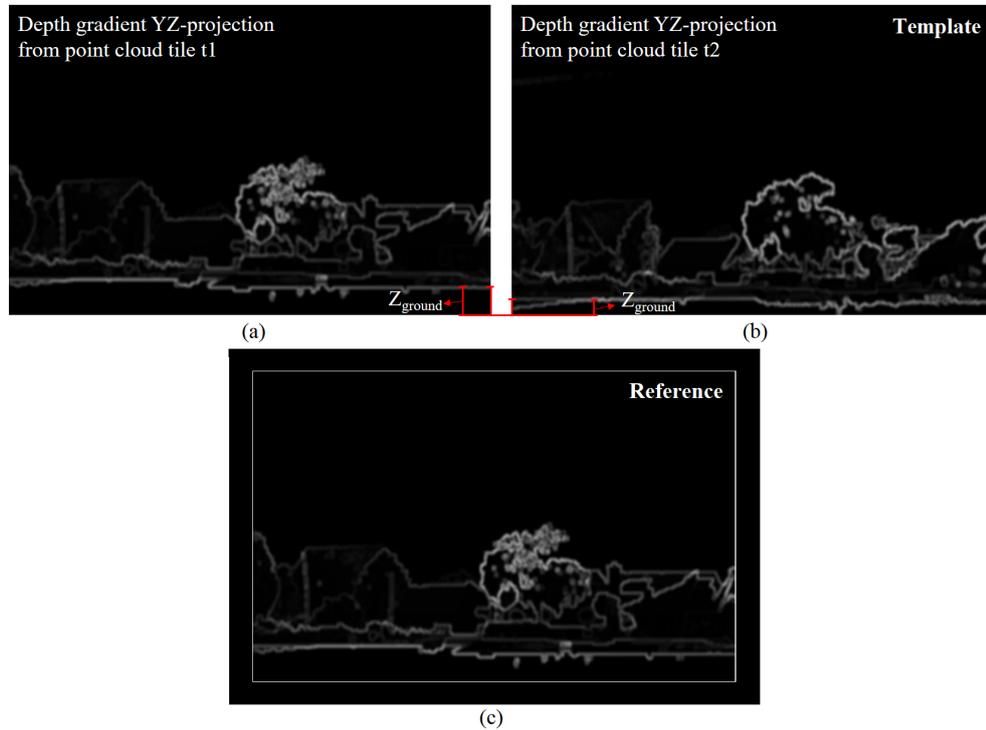
There are different methods that can be performed in order to apply a template matching technique [Ding et al., 2001]. The employed template matching method is based on a simple cross correlation statistical analysis of the brightness values of the two images. Theoretically the template image shifts over every possible location of the reference image, pixel by pixel<sup>1</sup>. At every location, a degree of similarity between the two images is calculated as the sum of all the multiplications between the corresponding pixels of the template and the reference image [Ding et al., 2001]. A cross correlation method is sensitive to possible differences in the brightness values of corresponding pixels between the two images that are matched [Sarvaiya et al., 2009]. Such differences may exist in the created images as explained. For example, the density values at corresponding pixels of an image pair may be different due to the different recording positions of the point cloud tiles. Despite this drawback of the cross correlation method, it is used in this project because high correlations can still be obtained if the pixel values follow the same pattern [Ding et al., 2001]. In other words, if the pixels' values increase and decrease the same way but the pixels' values are not completely the same, it is still possible to get a correct registration solution. Additionally, the method performs successfully even when small rotations and scaling is presented [Sarvaiya et al., 2009]. Since there might be some small rotations in the point clouds, as explained in Section 1.1.2, this property of the cross correlation method could be beneficial.

**METHOD** As a template matching technique determines the location of one image into another one, it is implied that the one image should be smaller than the other image. If the images in a pair have the same size, then there is nothing to compute. The conversion from 3D to 2D is conducted in this project in such a way that the images resulted from one overlapping pair have the same size (as described in page 48). In order to allow to the template matching technique to detect the matching position of the two images in a pair, search space must be created between the template image and the reference image. Therefore, a border of additional pixels is added to the reference image. The added pixels contain zero values. An ideal selection for the value of the added pixels would only positively influence the template matching process. Adding zero values was an easy decision in order to achieve the goal of creating search space between the template and the reference image. The number of the extra pixels is determined according to the maximum expected translation error and the grid cell size. The assumption that in worst case scenario the translation errors between the point clouds can be 5m is made. Such a big error is chosen for the cases where there is no GNSS reception at all, and the system is completely dependant on the IMU. For example, if the grid cell size of the created imagery equals to 0.05m, then approximately 100 grid cells (pixels) are added in each side of the image used as reference. The number 100 results by dividing 5m by 0.05m. Figure 4.26 illustrates two images created from two overlapping point clouds. This Figure indicates the reason why one of the images must be modified in order to be used as the reference image.

In the theory, it was mentioned that usually registration methods take into account a fixed and a moving scan (Section 2.1.2). The moving scan is moved to match the fixed scan. The cross correlation template matching technique in this

<sup>1</sup> The template matching technique constitutes theoretically a convolution of matrices in the spatial domain. The convolution in the spatial domain is usually implemented as a multiplication in the frequency domain through the Fast Fourier Transform, because it is a significantly faster than the convolution in the spatial domain. The OpenCV library which is used for the implementation of the template matching technique, performs the convolution in the frequency domain.

project is implemented twice. The first time one image from a pair is considered the reference or in other words the fixed image, and the next time the other. This is decided because there is no knowledge with regard to which point cloud set in a pair has the most correct absolute position, or even if any of them has. This is explained further in Section 2.1.2.



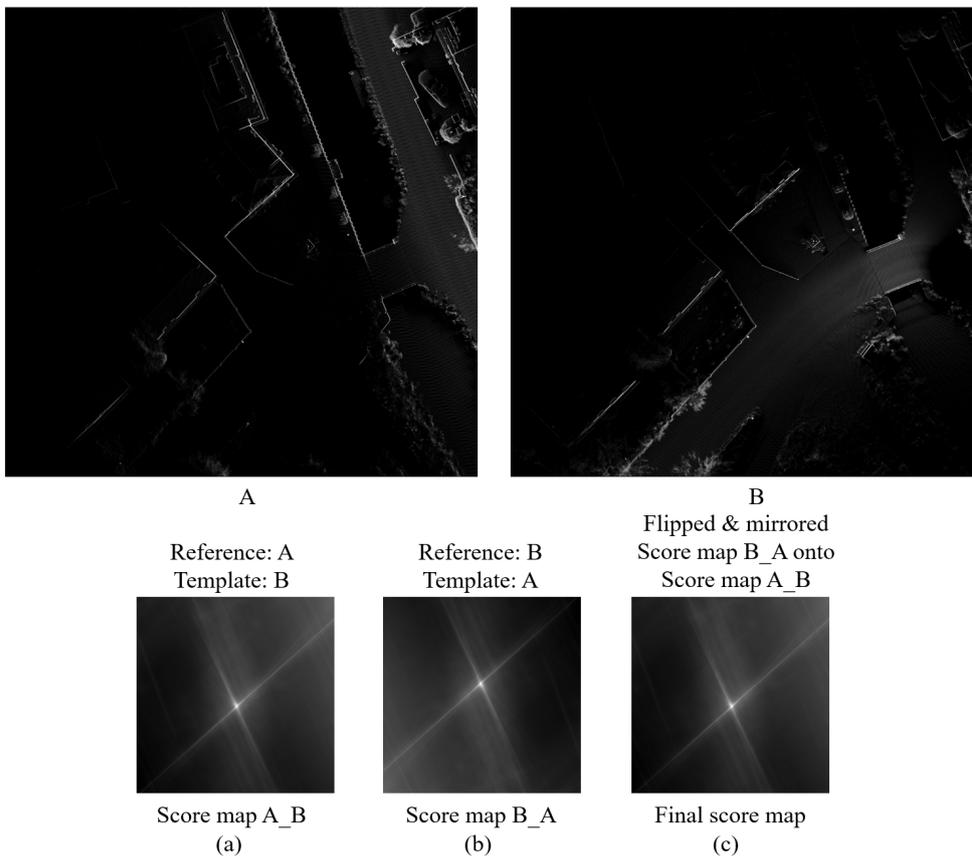
**Figure 4.26:** a) and b) Two depth gradient YZ-projections created from two overlapping point clouds. The images are created with the proposed method thus they have equal size. If these two images are superimposed, corresponding objects will not match. This can be seen by observing at the lines that represent the ground in the two images. There is clearly a different in the Z coordinates of the images. c) In order to allow to the used method to detect the position of the one image into the other, zero-pixels are added at each side of the image in (a). In such a way, (c) can be used as the reference image and (b) as the template. The white line in (c) is added here to distinguish the added pixels from the initial image.

Every time that the template matching is applied to match two images of the same type, the so-called score map is extracted. As described (in Section 2.5.5), a score map is a 2D array that contains the similarity scores computed between the template and reference image at all the possible overlay positions.

As the matching of each image pair is applied twice with swapped roles between the reference and the template image, one of the two resulted score maps is flipped horizontally and vertically. Then the modified score map is superimposed onto the other score map to retrieve the final output. A relevant example is illustrated in Figure 4.27.

The next step is to detect the image coordinates of the array cell in the score map that contains the highest similarity value. As explained and illustrated in the theory, the location of the highest similarity value is the location of the template image into the reference image. The highest similarity value is determined by detecting the maximum value among all the pixel values in the score map.

To summarize, let's take an example where a density XY-projection created from a point cloud tile 1 is matched with another density XY-projection which is created from a point cloud 2. Firstly, the template matching is performed by using the one XY-projection as reference and then, another template matching is performed by using the other XY-projection as reference image. The resulted score maps are



**Figure 4.27:** The two images on the top (A and B) are examples of two overlapping density XY-projections. a) The score map resulted from using the image A as reference and the image B as template during the matching process. b) The score map resulted from using the image B as reference and the image A as template during the matching process. c) The final score map resulted from superimposing the mirrored and flipped score map (b) on the score map (a).

combined in order to have one score map. The location of the highest pixel value is detected on the finalized score map. This 2D location is the location in the reference image, where the left top corner of the template image must be placed. Then the images will be matched.

Each 2D matching location which is defined with pixel units, is then used to determine the translation that registers the point clouds.

## 4.6 IMAGE COORDINATES TO SPACE COORDINATES

Since the developed method suggests the matching of images to ultimately match point clouds, the matching location of the images given in pixel units must be converted to meters.

**METHOD** It is important to understand that a matching location indicates the transformation parameters only in the directions that a plane is projected. If for example two XY-projections are matched, then the matching location indicates the translation parameters in X and Y axis only. If two XZ-projections are matched, then the matching location indicates the translation parameters in X and Z axis only. The similar applies when matching YZ-projections.

The coordinate represented in the width-axis of a projection is symbolized here with  $a$ . The coordinate represented in the height-axis is symbolized with  $b$ . For

instance, in an XY-projection  $X$  is represented by  $a$  and  $Y$  is represented by  $b$ . The matching location resulted from matching two projections of common visual information is defined by the location's width and height in pixel units. The width indicates the offset in the direction symbolized with  $a$  and the height indicates the offset in the direction symbolized with  $b$ .

$$matchLocation_{ab} = match_{width}, match_{height}$$

where  $a$ : the coordinate represented in the width-axis of a projection

$b$ : the coordinate represented in the height-axis of a projection

As explained in page 48 the creation of the projections from the point cloud tiles requires that the points' coordinates on the vertical axis acquire a negative sign. Consequently, a negative sign is assigned to the height component of the matching location so as to retrieve the correct offset in the vertical axis. Apart from this design decision, a number of pixels is added in each side of the image that is used as reference. The additional pixels are determined based on the expected translation error. The extra pixels must be subtracted from the matching location. To compute the amount of additional pixels, the expected error in meters added in one side of the image is divided by the grid cell size. As a result, the translation parameters in pixel units that are free from the design decisions become known. Lastly, the pixel units are converted in meters by multiplying with the grid cell size. To summarize, the calculation of the 3D transformation parameters is given by:

$$b_{translation} = -(match_{height} - ExpectedError / GridCellSize) * GridCellSize$$

$$a_{translation} = (match_{width} - ExpectedError / GridCellSize) * GridCellSize$$

where  $GridCellSize$ : is the width (which is equal to the height) of the created projections' cells in meters

$ExpectedError$ : the expected translation error in meters added in a side of the image

## 4.7 OPTIMAL SOLUTION & QUALITY EVALUATION

In this chapter, it is explained how the final transformation parameters are retrieved as they are computed many times from the multi-registration process. At the same time it is explained how the quality of the results is judged. The suggested method constitutes a combination of:

- Theoretical information from the Adjustment and Testing theory developed for traditional [geodetic observations](#) [Teunissen, 2000], [Sweco Nederland B.V., 2016]. Actual readjustment of the theory for [LiDAR](#) data is not conducted.
- Image processing techniques.
- Statistical measures.

This chapter reveals also the usefulness of creating various 2D projections from each 3D point cloud tile.

In the theoretical background, in Section 2.3, the term quality is explained with the reliability and precision. General definitions are given, but also definitions from the Adjustment and Testing theory for [geodetic observations](#) [Teunissen, 2000]. The given definitions are adjusted in this chapter for the local registration of mobile laser scanned point clouds. By doing so, the suggested method for the retrieval of the optimal solutions which is analyzed at the following steps, is justified.

**RELIABILITY AND PRECISION OF LOCAL POINT CLOUD REGISTRATION** The reliability and precision of the final optimum **geodetic observations** are based on the collected relevant redundant observations [Teunissen, 2000]. For this project, the observations taken into account at each local registration are the two overlapping point cloud tiles with the points included in each tile. Therefore, the maximum possible repetition of observations is two. Repetition is needed to calculate precision. Repetition could only exist at the overlapping area between the two point cloud tiles. After the application of the proposed registration method, a **MSE** could be calculated between the points at the overlapping area of each point cloud pair. However, it would be necessary first to detect the overlapping area between the point cloud tiles. Moreover, there is no redundancy which could be utilized to determine the internal reliability of the local registration results. In order to have redundancy, a mathematical relationship must exist between the observations to validate their correctness. For the geodetic measurements redundancy can be obtained when measuring all the angles of a triangle instead of measuring two and calculating the third. Nevertheless, a pair of overlapping point cloud tiles does not provide any relationship which could be used to validate the reliability of the relative registration results.

Instead of using the input data, namely the point cloud tiles, in order to judge the quality of the results, the estimated transformation parameters could be used. As the transformation parameters are computed with many different methods, they can be used as observations in order to judge the reliability and precision of the final optimum estimations.

By merging all the information given here and in the theory, and by taking into account the developed method up to now, the **quality of a relative registration between two point cloud tiles** is defined as:

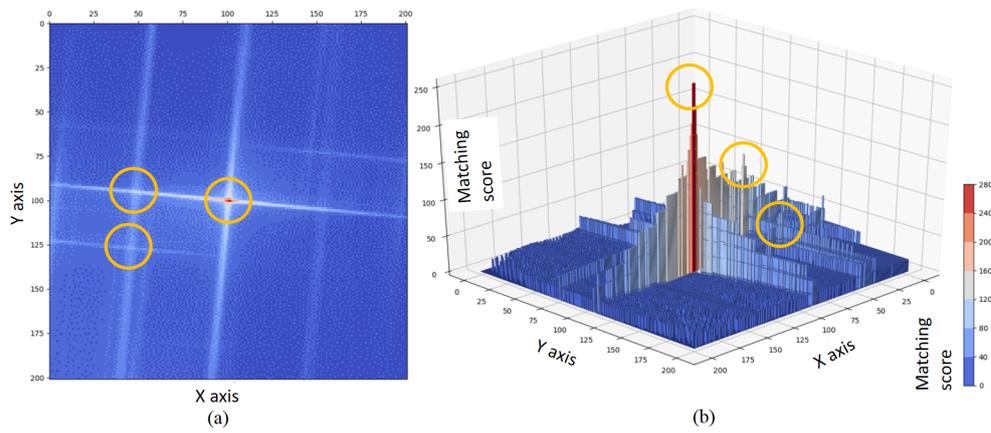
- A number that accompanies each final  $X$ ,  $Y$ ,  $Z$  translation parameter and indicates **the degree to which the parameter can be trusted**. Particularly, that number must describe how reliable and how precise each final translation is. The final adjusted output consists of a set of three final translations ( $X$ ,  $Y$ ,  $Z$ ) originating from multiple local registration results of the same **point-cloud pair**, retrieved from different methods.
- The **reliability** is defined as the extend to which errors are detected and how sensitive are the final estimations to undetected errors. These two are expressed by the internal and external reliability, respectively.
- The **internal reliability** determines how well errors are detected and thus, whether a translation parameter is strongly checked. It can be defined through a redundancy number. Redundancy is achieved when the computed redundant relative estimations obey to some mathematical rules. The higher the existing redundancy, the more powerful a measurement is. Also, the internal reliability of an estimation can be defined by the size of the smallest possible error that the estimation has. If the smallest possible observation error is small, then the measurement is strongly checked.
- The **external reliability** of a final transformation parameter determines the influence of a possible undetected error from a single registration estimation. Final transformations that are highly influenced by undetected errors have poor external reliability.
- The **precision** must be expressed by the repetition of local registration estimations taken into account for the computation of the final translation parameters. The smallest the variability in the considered estimations, the more precise the final estimation.

**ESTIMATION OF OPTIMAL SOLUTION - METHOD** The translation parameters are computed from nine different techniques, since nine image types are employed for the registration of each point cloud pair. Moreover, the transformation parameters are extracted twice when registering one image type because of the three different ways that the point cloud tiles are projected. Therefore a method to retrieve a single optimum X, Y and Z translation for every point cloud pair is required. To do so, the following steps are performed.

To obtain one optimum solution, firstly the results of low quality must be rejected. As the quality is expressed partly with the internal reliability which determines how well errors are detected, the first step is to use a method for errors detection. In other words, the aim is to detect the transformation parameters that might have been estimated incorrectly from the multi-registration method. The method must be capable of judging a single local transformation estimation individually. This is done by applying three evaluation criteria on the resulted score maps. The information that the score maps contain indicates the strength of the matching and that is the reason why the focus of the evaluation is on the score maps. The evaluation criteria are explained below.

1. **Criterion 1: The highest similarity value of a score map must be higher than a specific value.** As specified by Lemmens [1988] a threshold value must be applied on the highest similarity values of the score maps to eliminate the values that are not high enough. The lower the similarity value the weaker the matching between the two images. Some experiments were performed with which it was observed that the highest similarity values, retrieved when matching the created imagery, were either very high or very low. Particularly, the high values had order of magnitude higher or equal to 3 and the low values had order of magnitude -1. The lower boundary is set to 1.00 so as to exclude all the matching results with highest similarity values of order of magnitude -1. These values are considered significantly low in comparison with rest of the highest similarity values, and therefore the selected threshold is 1.00.
2. **Criterion 2: The highest similarity value of a score map computed with two different methods must be at the same image location.** As explained on page 64, the highest similarity value of a score map is initially determined by searching for the maximum value among the pixel values of a score map. The highest similarity value is computed again at this step with a different method. This method employs a maximum filter<sup>2</sup> to detect the local maxima in the score maps. Local maxima constitute peak pixels formed by their surrounding pixel values. It is possible that in a score map there are more than one peaks. Some of those will have higher and some lower values (in score maps of good quality) or all of them will have high values (in score maps of bad quality). The detected peak pixels with the maximum filter method are sorted from the highest to the lowest value. The peak with the highest value constitutes the matching location computed with this method (Figure 4.28). The expectation is that the matching location returned from the first method (page 64) is the same as the matching location determined with the technique explained here. If this criterion is validated then there is no doubt that there is only one matching location. In such case a score map passes the present criterion.
3. **Criterion 3: If more than one peaks of similarity values exist in the score maps, then the matching location is accepted if its value is significantly higher than the value of the second highest peak.** To check the strength of

<sup>2</sup> A maximum filter employs a running window which is an array that slides over the score map image. At every location that the running window is shifted, a cell of the running window corresponds to a pixel in the score map. The pixel of the score map that corresponds to the central cell of the window is replaced by the brightest pixel among the pixels included in the window.



**Figure 4.28:** a) An example of a score map in 2D. b) The score map of (a) in 3D. The yellow circles indicate the local maxima in the specific score map, in 2D, and in 3D. The highest peak among all the three local maxima is found in the middle of the score map and is illustrated with red color.

the highest matching score, a comparison is made between the pixel values of the first and second highest peak, in case there are more than one local maxima. The technique used to detect the peaks of similarity values is described in the previous criterion. If there is a significant difference between the values of the highest two peaks, then the first highest peak is undoubtedly the best matching location. Specifically, the difference in the values of the two highest peaks is found and converted to a percentage. After conducting some experiments with the given data, it was decided that by setting a threshold of 3%, the unreliable score maps could be detected. If the difference between the values of the two highest peaks is higher than 3% then the highest peak can be highly trusted. If the difference between the values of the two highest peaks is lower than 3% then the highest peak is not trusted.

The score maps that are taken into account for the computation of the optimal transformation parameters are those which pass all the evaluation criteria. Even if a score map passes two out of the three criteria, it is still rejected. This decision is made to increase the possibility that only transformation parameters of good quality are considered for the final estimations. The score maps that pass all the evaluation criteria are taken into account to reduce the computed redundant solutions. Particularly, the translation parameters ( $X$ ,  $Y$ ,  $Z$ ) which are computed 27 times per point cloud pair are decreased to 9. To explain the method, the Table 4.3 is given. The row 0 of the table indicates the image type. The column 0 of the table initially indicates the projection type. The cells that are filled with a 'Yes' or 'No' show if the evaluation criteria are passed for a specific image type projected on a specific plane. For example, the cell in row 1 and column 2 shows that the matching of two depth  $XY$  projections has been accepted since all the evaluation criteria are passed. This means that the  $X$  and  $Y$  translation parameters computed from this specific image pair can be used for the computation of the optimal solutions. Also, the cell in row 2 and column 2 shows that the pair of  $XZ$  projections passed the assessment criteria. In comparison, the example shows that the matching of depth  $YZ$  projections did not pass the evaluation criteria (row 3, columns 2). As a result, there are two  $X$  parameters computed from the depth image type. One from the  $XY$  projection and one from the  $XZ$  projection. The  $X$  can be then computed for the depth image type as the average of the two  $X$  translations. In comparison, the  $Y$  parameter is only accepted from the  $XY$  projection. Thus, the  $Y$  parameter from the depth imagery is equal to the  $Y$  translation parameter as given from the matching of the  $XY$  image pair. Furthermore, it can be seen that in the example, none of

projections of the depth gradient has been accepted. Therefore, the depth gradient images do not contribute at all to the final estimations. Overall:

- The solutions produced from the multi-registration process of each point cloud pair are treated firstly per image type. The alternative way, would be just to treat together all the estimations of each translation which pass the evaluation criteria. The decision to merge the estimations firstly per image type is made because the projections of one image type contain the same information. This information is the specific attribute of the 3D points that the projections illustrate.
- Each parameter is computed from two projections per image type. If the matchings of both projection pairs (from which one translation parameter is computed) are accepted, then the X, Y and Z of the specific image type are computed as the average of the two solutions retrieved from the two projections' matchings.
- If the matching of only one projection pair among the two projection pairs (from which one translation parameter is computed) is accepted, then the X, Y and Z resulted from a specific image type are equal to the one solution from the one projection's matching.
- If the matchings of both projection pairs (from which one translation parameter is computed) are rejected, then the two translation parameters computed from the specific image type are not considered for the computation of the final solution.

	0	1	2	3	4	5	6	7	8	9
0		density	depth	depth gradient	...					
1	XY projection	Yes	Yes	No						
2	XZ projection	Yes	Yes	No						
3	YZ projection	Yes	No	No						
4	<b>X translation from an image type, <math>\bar{X}</math></b>	$(X_{xy} + X_{xz}) / 2$	$(X_{xy} + X_{xz}) / 2$	-						
5	<b>Y translation from an image type, <math>\bar{Y}</math></b>	$(Y_{yz} + Y_{xy}) / 2$	$Y_{xy}$	-						
6	<b>Z translation from an image type, <math>\bar{Z}</math></b>	$(Z_{yz} + Z_{xz}) / 2$	$(Z_{yz} + Z_{xz}) / 2$	-						

Table 4.3: The table shows the redundant solutions are decreased from 27 per translation parameter (X, Y, Z) to 9 per translation parameter.

The optimum X, Y and Z translation parameter are the average of the estimations resulted from the previous step. For instance, the optimal X solution of the example given in the Table 4.3 is the average of all the values included in the cells 4,1 till 4,9 where the first number indicates the row and the second indicates the column. The maximum amount of estimations that can be taken into account for the optimal solution's calculation is nine, as the image types are nine. Nevertheless, the included solutions can be less than nine, depending on the amount of accepted score maps per point cloud pair.

**INTERNAL, EXTERNAL RELIABILITY AND PRECISION - METHOD** The Figure 4.29 illustrates an overview of the method followed for the quality evaluation of the results.

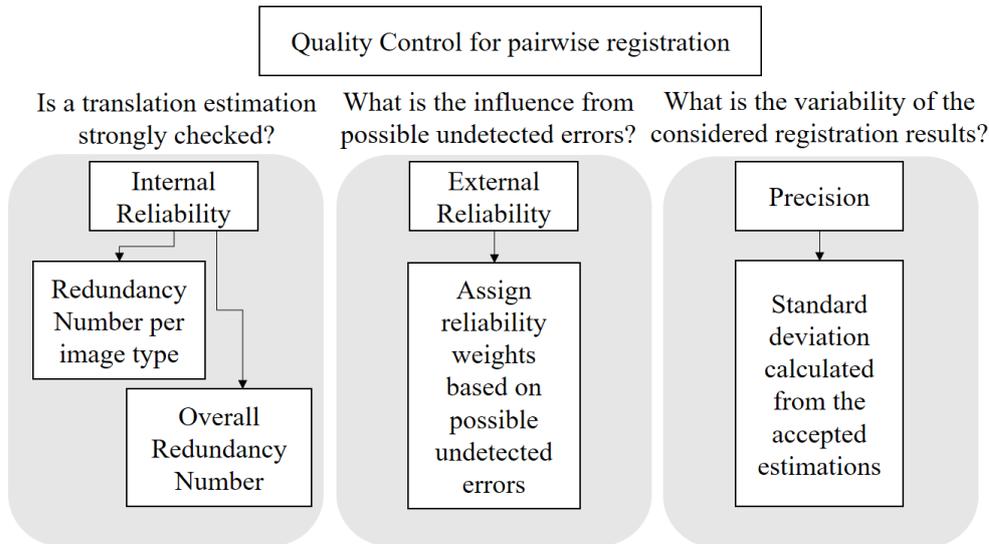


Figure 4.29: Overview of the method followed for the quality evaluation of the results.

In order to provide insight about the **internal reliability** of the results it must be checked how well errors are detected, or in other words, whether a translation estimation is strongly checked. An indication about the internal reliability is given:

- for the estimations retrieved from the registrations of every image type and
- for the final optimum estimation.

The ultimate purpose is to judge the internal reliability of the optimal solutions. However, the optimal solutions are based on the estimations retrieved from the matchings of the different image-types. Thus, firstly the internal reliability of those is evaluated. A mathematical relationship between the computed translation parameters is required. The only correlation that can be claimed is the equality between the translation parameters retrieved from the registrations of the same image-type, but from different projections. For instance, the X translation estimated from the registration of the density XY-projections and the X translation estimated from the registration of the density XZ-projections, must be the same. This kind of correlation can be assumed since the registration is performed with the same type of information. Namely, both projections visualize the same attribute of the points.

Redundancy exists when the matchings of both projection pairs (from which one translation parameter is computed) are accepted, and the estimations from the matchings are equal. Otherwise, redundancy does not really exist and the estimation is internally not reliable. However, here redundancy is considered the amount of estimations from a specific image-type that are taken into account for the calculation of the optimum solution. This modification is made so as to gain detailed insight on how strong an estimation from a specific image-type is. Below this concept is explained and illustrated in Table 4.4.

1. If the matchings of both projection pairs (from which one translation parameter is computed) are rejected, then the redundancy number of the translation parameter estimated from the specific image-type is equal to zero out of two (0/2). For example, in Table 4.4 the cell 7,3 (row, column) shows that the internal reliability of the X translation parameter from the depth gradient images is 0/2. This is because the estimation of the X translation

parameter from the depth gradient projections does not exist. The estimation does not exist because the score maps of the XY and XZ projection did not pass the evaluation criteria (see Table 4.3).

2. If the matching of only one projection pair among the two projection pairs (from which one translation parameter is computed) is accepted, then the redundancy number of the translation parameter estimated from the specific image-type is equal to one out of two (1/2). For example the cell 8,2 shows that the internal reliability of the Y translation parameter computed from the depth images is equal to 1/2. This is because the Y parameter of the depth images is retrieved only from the matching of the depth XY projections. The matching of the depth YZ projections is rejected.
3. If the matchings of both projection pairs (from which one translation parameter is computed) are accepted, then the redundancy number of the translation parameter estimated from the specific image-type is equal to two out of two (2/2).
4. If the matchings of both projection pairs (from which one translation parameter is computed) are accepted and they are equal, then the redundancy number of the translation parameter estimated from the specific image-type is symbolized with a two plus out of 2 (2<sup>+</sup>/2). This is the highest possible redundancy that can be retrieved from the registrations of one image type.

	0	1	2	3	4	5	6	7	8	9
0		density	depth	depth gradient	...					
1	$\bar{X}$	$(X_{xy} + X_{xz}) / 2$	$(X_{xy} + X_{xz}) / 2$	-						
2	$\bar{Y}$	$(Y_{yz} + Y_{xy}) / 2$	$Y_{xy}$	-						
3	$\bar{Z}$	$(Z_{yz} + Z_{xz}) / 2$	$(Z_{yz} + Z_{xz}) / 2$	-						
4	Internal reliability X (Redundancy number)	2/2	2/2	0/2						
5	Internal reliability Y (Redundancy number)	2/2	1/2	0/2						
6	Internal reliability Z (Redundancy number)	2/2	2/2	0/2						

Table 4.4: The table is an extension of the example shown in Table 4.3. It illustrates the internal reliabilities of the estimations outputted from each image type.

To determine the internal reliability of the X, Y and Z optimum solution, simply the amount of image-types included for computation of the final estimations are counted. To illustrate this, the Table 4.5 is given. Each cell of the row 1 shows the amount of matchings that passed the evaluation criteria and thus they are taken into account for the computation of the final X translation parameter. Since only the depth gradient images (cell 1,3) does not contribute to the final X estimation, the

overall redundancy is equal to 8/9. Nine is the amount of image types that are used in total for the registration process.

	0	1	2	3	4	5	6	7	8	9	10
0	Image type	den- sity	depth	depth grad.	...						Ove- rall RN
1	Internal reliability X (RN)	2/2	2/2	0/2	2/2	2/2	2/2	2/2	1/2	2/2	8/9
2	Internal reliability Y (RN)	2/2	1/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	4/9
3	Internal reliability Z (RN)	2/2	2/2	0/2	2/2	2/2	1/2	1/2	0/2	1/2	7/9

**Table 4.5:** The table is an extension of the example shown in Table 4.3. It illustrates the overall internal reliability of each estimation.

In order to provide insight about the **external reliability** of the final estimations, information about the influence of undetected errors must be provided. Errors that might not be detected with the explained evaluation criteria (presented in page 68) could take place when:

1. there are changes in one scene which is captured more than one times. For instance, the first time that a scene is captured, a car is parked in front of a building, but the second time the scene is captured, the car is not any longer parked.
2. features with similar geometry, but different elevations are matched. For example, two point clouds from the same area might include a road which has a bridge on the top. Since a road and a bridge could possibly have similar geometry, they could possibly be matched.
3. the observation point of the recording vehicle was far away from the scanned scene. That means that only a few points recorded and especially less than when the car was recording from a short distance. In such cases, the registration process could be hindered as the overlap between two point clouds may not be existed or the overlap between them may be very small.

In this project, only the last cause of possible undetected errors is employed to provide information about possible influence on the final estimations. If few 3D points scanned in a point cloud tile, then there will be many empty pixels in the corresponding constructed projection. Empty pixels are those that are illustrated with black. The employed method accumulates the pixels that are not black, namely the pixels that include information from 3D points. According to the pixel's size, the area in  $m^2$  that those pixels cover, is calculated. Then the percentage of that area is calculated by taking into account the total amount of pixels in the image, and the area that the image covers in  $m^2$ . This percentage is calculated for both images in an image pair. It can be the case that one of the images represents a point cloud tile with many points and the other image represents a point cloud tile with a few points. Thus, in the one image there are will be many filled pixels, while in the other many pixels that are empty. Such cases could hinder the registration process as the overlap between them could not exist or the overlap between them could be small. Therefore, in every case the minimum percentage from the two involved images is

chosen. According to that percentage, weights that indicate how (externally) reliable a matching might be, are given. These weights are illustrated in Table 4.6. For example, the percentage of the area covered by non-empty pixels is computed for two images that contain common visual information. If the least percentage among the two is equal or higher than 40%, then it is assumed that the external reliability of the specific matching is 0.9. A percentage of minimum 40% covered area is considered high, therefore the external reliability assigned is high (0.9). The external reliability weights are related only to the explained possible cause of errors.

External Reliability (ER)	Least % of covered area
0.9	$\geq 40\%$
0.8	$\geq 20\%$ AND $< 40\%$
0.6	$\geq 4\%$ AND $< 20\%$
0.4	$\geq 1\%$ AND $< 4\%$
0.1	$\geq 0\%$ AND $< 1\%$

**Table 4.6:** The table shows the (external) reliability weights given to image registration results according to the image area ( $m^2$ ) that adopts information from the 3D points.

In order to provide insight about the **precision** of the final estimations a standard deviation is computed. The standard deviation shows the variability of the estimations resulted from each image-type registrations. The equation 4.2 indicates how the standard deviation is computed per final translation parameter. The Table 4.7, which is an extension of the example shown in Table 4.3, shows in detail how the standard deviations are computed.

$$Ov.\sigma_{tp} = \sqrt{\frac{\sum_{i=1}^n (\bar{tp}_i - Ov.\bar{tp})^2}{n - r_{tp}}} \quad (4.2)$$

where

- $Ov.\sigma_{tp}$  : Overall standard deviation of a final translation parameter. X, Y and Z are the translation parameters.
- $i$  : It refers to the image type from which a mean translation parameter is retrieved. For example, density image matchings, depth, depth gradient etc.
- $n$  : It is the total amount of image types from which a mean translation parameter can be computed. This equals to 9, since 9 image types are created per point cloud tile.
- $\bar{tp}$  : The (mean) translation parameter computed from the registrations of the two projections pairs of an image type.
- $Ov.\bar{tp}$  : The mean translation parameter from all the translation parameters retrieved per image type matchings, which pass the evaluation criteria.
- $r_{tp}$  : The amount of the image types from which both translation parameters are rejected, when the evaluation criteria are applied.

	0	1	2	3	4	5
0		density	depth	depth grad	Other image types ..	Overall mean / Overall stand deviation
1	$\bar{X}_i$	$\frac{(X_{xy} + X_{xz})}{/2}$	$\frac{(X_{xy} + X_{xz})}{/2}$	-		$Ov.\bar{X} = \frac{\sum_{i=1}^N (\bar{X}_i)}{9 - r_x}$
2	$Ov.\sigma_X$					$Ov.\sigma_X = \sqrt{\frac{\sum_{i=1}^9 (\bar{X}_i - Ov.\bar{X})^2}{9 - r_x}}$
3	$\bar{Y}_i$	$\frac{(Y_{yz} + Y_{xy})}{/2}$	$Y_{xy}$	-		$Ov.\bar{Y} = \frac{\sum_{i=1}^9 (\bar{Y}_i)}{9 - r_y}$
4	$Ov.\sigma_Y$					$Ov.\sigma_Y = \sqrt{\frac{\sum_{i=1}^9 (\bar{Y}_i - Ov.\bar{Y})^2}{9 - r_y}}$
5	$\bar{Z}_i$	$\frac{(Z_{yz} + Z_{xz})}{/2}$	$\frac{(Z_{yz} + Z_{xz})}{/2}$	-		$Ov.\bar{Z} = \frac{\sum_{i=1}^9 (\bar{Z}_i)}{9 - r_z}$
6	$Ov.\sigma_Z$					$Ov.\sigma_Z = \sqrt{\frac{\sum_{i=1}^9 (\bar{Z}_i - Ov.\bar{Z})^2}{9 - r_z}}$

**Table 4.7:** The table is an extension of the example given in the table 4.3. It shows how the precision is computed for each final translation parameter. The final translation parameters are symbolized as  $Ov.\bar{X}$ ,  $Ov.\bar{Y}$ ,  $Ov.\bar{Z}$ . The precision of the final parameters are based on the computation of the standard deviation. The standard deviations are symbolized as  $Ov.\sigma_X$ ,  $Ov.\sigma_Y$  and  $Ov.\sigma_Z$ , and they are computed from the accepted solutions retrieved per image type. If an image type does not provide any solutions, then only the remaining solutions are considered.

To summarize, the quality of the final registration parameters is indicated with three numbers. These numbers define the internal reliability, the external reliability and the precision of each final parameter. Moreover, the translation parameters computed from each image-type are accompanied with a number that shows their internal reliability. In other words, it shows how strongly each parameter is checked.

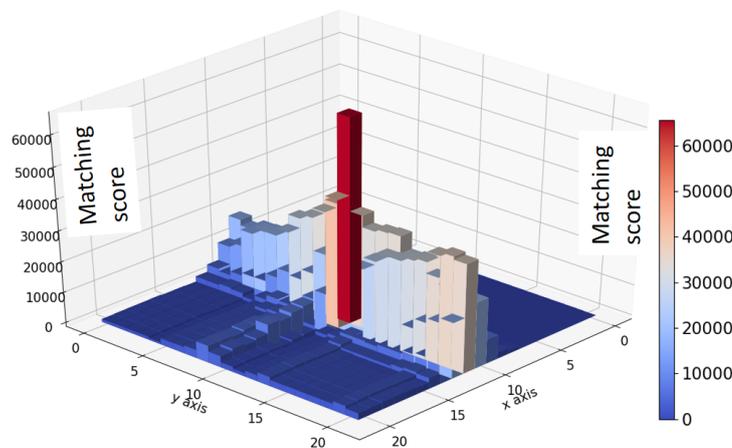
## 4.8 SUB-PIXEL ACCURACY

The accuracy of the results with the proposed method can not be better than the accuracy of the images' grid cell size. That is because the method detects the pixel where the best matching between two images is found. As a result, the accuracy of the translation parameters is always equal to the resolution of the images. For example if the images' pixel size is 5 cm, then it is not possible to detect a potential translation error of less than 5 cm. To gain more insight about the problem, a score map resulted from the registration of two images is visualized in 3D and magnified 4.30. One can see that the result of the matching is not a single point, but is a whole square (that is the upper face of the bar that represents the peak). That implies that the matching solution lays somewhere in that square, or in other words, lays somewhere in the pixel that indicates the matching location.

It is important to clarify that since the optimum translation parameter is calculated as the average from the accepted estimations, the final estimation could possibly

have sub-pixel accuracy. But this accuracy is retrieved by calculating the average translation and it is not computed from the registration method.

Apart from the fact that by performing a sub-pixel accuracy method the results will have better accuracy, there is another reason why a sub-pixel accuracy method would be beneficial. The execution time of the developed algorithm is related with the grid cell size of the created images. That is because the size of the grid cells determines the amount of cells in an image. The computation time of the developed algorithm would be less if a large grid cell size is used. The bigger the grid cell size, the less the grid cells of an image. As the cells are less when large grid cell size is used, it means that the binning of 3D points into 2D grid cells will be performed less times. It also means that the template matching will have to examine less pixels. Nevertheless, when large grid cells are used, the accuracy of the translation parameters will be very poor. However, if a simple sub-pixel accuracy method is applied on the registration results retrieved by constructing images with large grid cells, then the results will obtain high accuracy. Moreover, the execution time would be less than when using small grid cell size. For that reason, a sub-pixel accuracy method is wanted.



**Figure 4.30:** a) An example of a score map image visualized in 3D. The score map illustrates the similarity scores between two images that are matched. The highest score is colored with red. The emphasis of the Figure is on the fact that the matching solution does not have point-resolution, but a pixel-resolution. (The score map used to create the graph has grid cell size 0.5m. By using a score map with low resolution, the problem of having discrete matching values is visually stressed).

**METHODOLOGY** Lemmens [1992] and Zhang et al. [2009] interpolate over the discrete values of an image in order to obtain sub-pixel accuracy. Thus, the main idea of the developed method is to apply an interpolation method on the discrete pixel values of the score maps. The interpolation method is performed on the score map arrays because they include the matching information. The pixel with the highest similarity indicates the matching location. By applying an interpolation method, the exact point where the highest similarity score lays, can be found. The sub-pixel accuracy method is only applied to the score maps that pass the quality evaluation criteria (see page 68). Since the peak pixel is computed earlier with two methods (see pages 64 and 68), the interpolation is specifically applied over the peak pixel value and its neighbouring pixels in a score map (Figure 4.31). The challenging part is to find what kind of interpolation method to use so as to describe best the way the pixel values are distributed. In other words, the function that could be best fitted in the area over the peak pixel is searched.

Theoretically it would be logical that the pixel values over the peak pixel in the score maps are distributed almost normally (Gaussian distribution). As mentioned

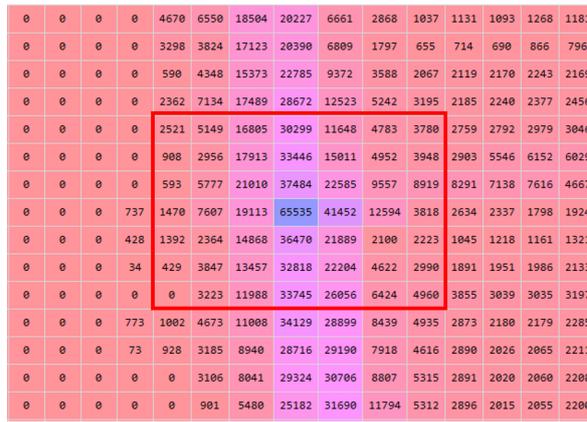


Figure 4.31: The area on a score map where a sub-pixel accuracy method may be applied. The blue pixel indicates the highest similarity value. The red square includes the neighboring pixels to the peak pixel. The square includes 3 neighboring pixels in each direction. This could be different but for this visualization 3 neighbors have been drawn. (The example score map is the output from images with the grid cell size was 0.5m x 0.5m. For visualization purposes, some columns and rows at the edges of the score map have been cropped).

by Naidu and Fisher [1991], the light incident on the laser scanner sensor is distributed nearly normally. Thus, the points in a small area, such as the area covered by a small pixel, are normally distributed. However, the developed method bins the points in pixels and provides one value per pixel. Thus, the Gaussian distribution begins to fade away. Furthermore, the template matching method is applied and the similarity values are computed. Therefore, the way the values distributed in a score map deviates more from a Gaussian distribution.

Moreover, it is considered that the distribution of the values around the highest similarity value is not symmetric. This is believed because the scanned environments are mostly not symmetric, and even more rarely are symmetric in every direction. In such cases, the similarity values that surround the peak value would deviate in different directions, such as in vertical and horizontal direction. The similarity values could even deviate when they are in the same direction, but in opposite orientations, for example on the left and the right of the peak value.

Firstly, a polynomial fitting was conducted. The equation that describes the fitted polynomial can be used as an interpolation method to find the highest similarity value with sub-pixel accuracy. Lemmens [1992] suggests two relevant methods. The 2D discrete values of the score maps can be approximated with two 1D second order polynomials. The two 1D polynomials are fitted on the horizontal and vertical lines of pixels that intersect on the highest similarity value (Figure 4.32).

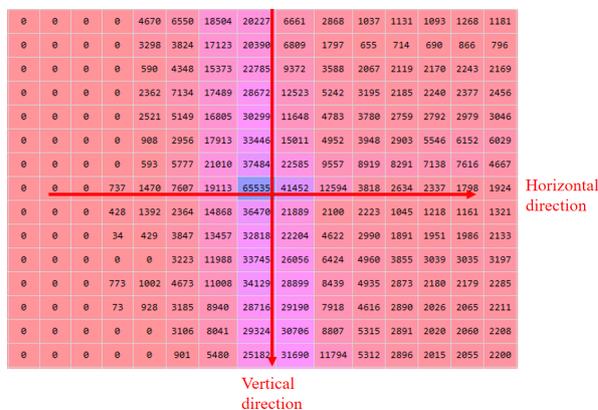
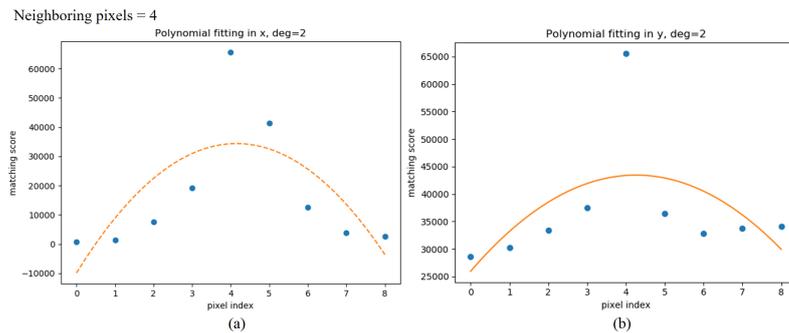


Figure 4.32: The Figure illustrates where exactly a horizontal and a vertical curve is fitted on the score maps in order to interpolate the discrete pixel values.

The other method suggested by Lemmens [1992] considers the fitting of a curved surface, described by a second order polynomial. In general, it is considered a better solution to fit a surface instead of two 1D curves. By fitting a surface, the whole neighbourhood is taken into account. However, the fitting of two 1D curves is a simplified solution. In any case, both methods suggested by Lemmens [1992], employ second order polynomials. To test if a second order polynomial is a good way to model the values in the score maps, two 1D second order polynomial curves are fitted horizontally and vertically. Score maps resulted from low resolution images have been used to test the method. Such score maps contain less values than the score maps resulted from high resolution images. In other words, the similarity values are more sparse than in the score maps resulted from high resolution images. It is more challenging to fit a function on sparse values, than on very dense values. If the developed method succeeds with low resolution score maps then, there is high chance that the method will also be successful with high resolution score maps.

The Figure 4.33 shows a second order polynomial fitting on the values of the score map shown in Figure 4.32. The Figure 4.33a shows how a second order polynomial is fitted on the highest discrete similarity value and on 4 neighbouring pixel values from the peak's left and right side. The Figure 4.33b shows how a second order polynomial is fitted on the highest discrete similarity value and on 4 neighbouring pixel values from the peak's upper and bottom side. It can be seen that the fitted created curves deviate significantly from the pixel values. This indicates a non-representative fitting by the second order polynomial curves. Although, only one example of fitting a second order polynomial curve is shown here, some other experiments have shown the same insufficient fitting results.



**Figure 4.33:** a) Horizontal 1D second order polynomial fitting over the peak pixel of the score map shown in Figure 4.32 and 4 neighbors of it in each side. b) Vertical 1D second order polynomial fitting over the peak pixel of the score map shown in Figure 4.32 and 4 neighbors of it in each side.

Before this option was totally excluded, the same method was performed by using only 2 neighbouring pixels. By having less observations, in other words less pixel values, the restrictions of the fitting are less. Thus, the fitting method is facilitated. Nevertheless, as it can be observed in Figure 4.34 the fitting results deviate significantly from the observations.

Since, the second order polynomial provided poor results, polynomials of higher order were examined. Score maps created with images of different resolutions have been used. Additionally, different numbers of neighbouring pixels have been considered. As the Figure 4.35 shows, the order of the polynomial which fits best the observations is in every case different. According to the input parameters, different orders of polynomials fit best to the score maps. However, as the Figure 4.36 shows, when the input parameters do not change, the same order polynomial fits best different score maps. The Figure shows the fitting result of only 3 score maps, but more were examined and the order of the polynomial fitted best to the observations was the same.

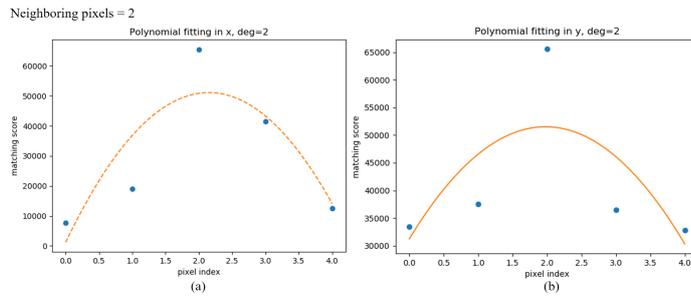


Figure 4.34: a) Horizontal 1D second order polynomial fitting over the peak pixel of the score map shown in Figure 4.32 and 2 neighbors of it in each side. b) Vertical 1D second order polynomial fitting over the peak pixel of the score map shown in Figure 4.32 and 2 neighbors of it in each side.

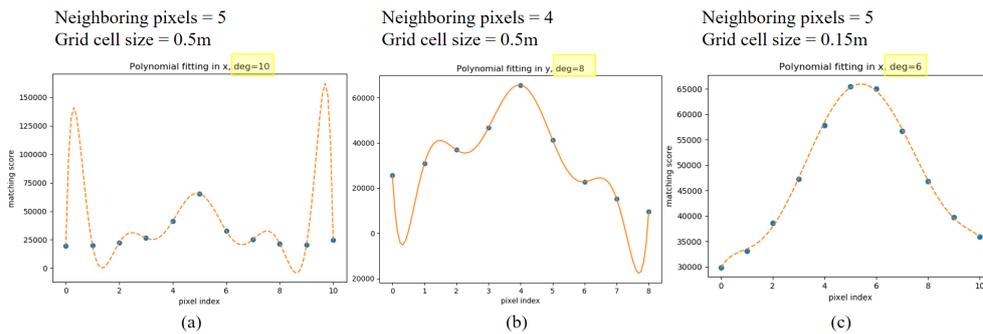


Figure 4.35: The Figures show the fitting of 1D curves on the values over the highest similarity pixel of different score maps. a) Five neighbouring pixels are considered. A polynomial of tenth order fitted successfully the horizontal values over the peak pixel value of a score map resulted from images' resolution 0.5m. b) Four neighbouring pixels are considered. A polynomial of eight order fitted successfully the vertical values over the peak pixel value of a score map resulted from images' resolution 0.5m. c) Five neighbouring pixels are considered. A polynomial of sixth order fitted successfully the horizontal values over the peak pixel value of a score map resulted from images' resolution 0.15m.

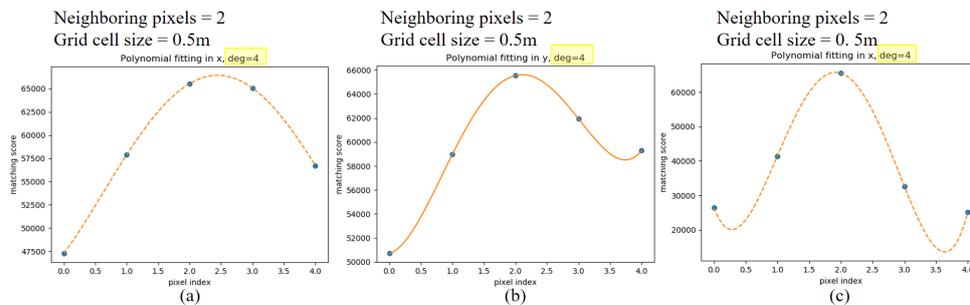
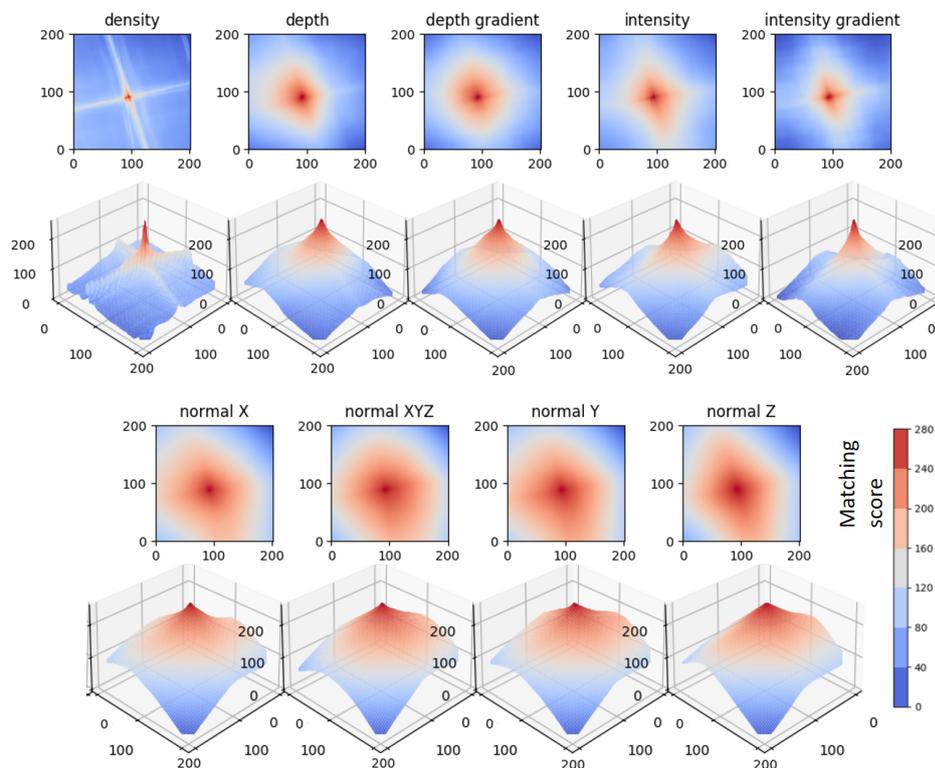


Figure 4.36: The Figures show the fitting of 1D curves on the values over the highest similarity pixel of different score maps. All the score maps used resulted from images' resolution 0.5m. Also in all the cases 2 neighbouring pixels are considered in both directions (vertical and horizontal). For all the cases a polynomial of fourth order fits best the pixel values over the peak value.

Based on the conducted research, the conclusion is that a polynomial fitting could be used if the input parameters are stable. The number of the neighboring pixels can be stable, but the grid cell size of the created images is something that changes according to the user's desire. A possible solution to this would be to make the algorithm examine different orders of polynomial. Then, the algorithm would choose the order of polynomial that gives MSE (between the pixel values and the corresponding points on the fitting line) equal or close to zero. However, such

technique would be a bit complex as for every grid cell size, different order polynomials would be required. This means that different functions must be used to find the peak of the polynomials<sup>3</sup>, according to the polynomial's order. For that reason, other methods are examined.

The next step constitutes the 3D visualization of several score map images that resulted from high image resolutions. The purpose is to observe if the score map values follow a specific distribution. Then this could be used as the interpolation method. The visualization is performed on high resolution images because those are constituted from dense values, thus the distributions are better described. Since the Velodyne HDL-32E LiDAR scanner from which the data used in this project has accuracy 2cm, high resolution score maps are considered those that have accuracy (grid cell size) from 2cm to 15cm. The Figure 4.37 shows examples of score maps (in 2D and in 3D) resulted from images with resolution 5cm. The creation of the 3D visualizations is made by extruding the matching scores. It is important to mention that the score maps shown in the Figure 4.37 constitute ideal examples, as the high similarity values are quite distinct; they are sharp and their values are high.



**Figure 4.37:** The score maps shown are outputs of the registration of different image pairs, created from the same point cloud tile pair. The first line includes score maps in 2D. The labels above the 2D score maps indicate the type of images matched to give the score map the is placed below. The second line illustrates the 3D visualization of the 2D score maps. The third line has more score maps in 2D and again the line below illustrates the corresponding score maps in 3D.

The conclusion that can be drawn by observing the (ideal) score maps is that the similarity values seem to be distributed with a 2D Gaussian or a 2D Laplacian function, which is also known as a Bivariate Laplacian function (Figure 4.38). Sometimes the peak of the score maps is not rounded. This can lead to the conclusion that the distribution of the values is closer to a 2D Laplacian than to a 2D Gaussian. Peaks that are not rounded are illustrated clearly on the score maps in 3D at the second line of the Figure 4.37. Despite this, when the 3D score maps are

<sup>3</sup> The peak of a polynomial corresponds to the highest similarity value between the two matched images, that has sub-pixel accuracy.

observed, it can be seen that the way the values are distributed at the edges does not remind either a Laplacian or a Gaussian function. Both functions are flat at the edges, while the score maps appeared to be a bit curved at the edges. Nevertheless, the Laplacian and the Gaussian are the two distributions that are closer to the distribution of the values in the score maps<sup>4</sup>.

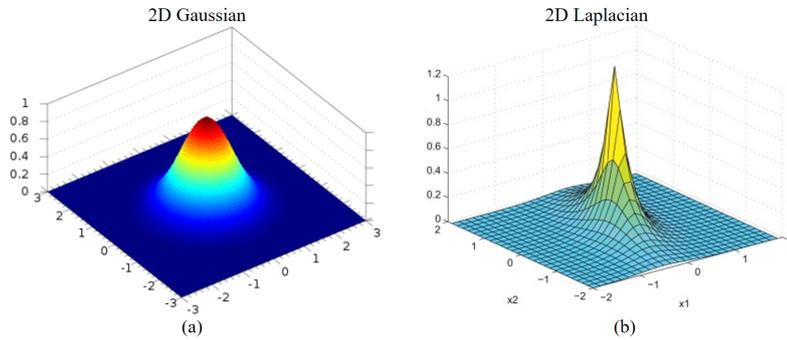


Figure 4.38: a) An example of a 2D Gaussian Probability Density Function. b) An example of a 2D Laplacian Probability Density Function

It is finally chosen to model the values over the peak pixel in the score maps with a 2D Gaussian function. The reasons are:

- There is no certainty about the perfect way to model the similarity values in the score maps. The score maps seem to be distributed with a function similar to a Gaussian or a Laplacian.
- It was explained that theoretically the values in the score maps are expected to be distributed with a function similar to a Gaussian (see page 76).
- The equation for a 2D Gaussian is simpler than that of a 2D Laplacian (Figure 4.39). A simpler equation could possibly lead to a less complex implementation. This is considered important as by applying a sub-pixel accuracy an additional step is added to the proposed method. This step aims to improve the accuracy and possibly the speed of the method. By applying a complex method, then there is a chance that the algorithm becomes more complex than it was, or it even needs longer execution time. These are both undesired.

- Asymmetric bivariate Laplace distribution, where  $k = 2$

$$f_{\mathbf{x}}(x_1, \dots, x_k) = \frac{2e^{\mathbf{x}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}}{(2\pi)^{k/2}|\boldsymbol{\Sigma}|^{0.5}} \left( \frac{\mathbf{x}'\boldsymbol{\Sigma}^{-1}\mathbf{x}}{2 + \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}} \right)^{v/2} K_v \left( \sqrt{(2 + \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})(\mathbf{x}'\boldsymbol{\Sigma}^{-1}\mathbf{x})} \right)$$

- 2D Elliptical Gaussian function

$$f(x, y) = A \exp \left( - \left( a(x - x_o)^2 + 2b(x - x_o)(y - y_o) + c(y - y_o)^2 \right) \right)$$

Figure 4.39: The upper equation describes an (asymmetric) bivariate Laplace function (Source [Kozubowski et al., 2013] and Wikipedia). The lower equation describes an (elliptical) 2D Gaussian function (Source [Anthony and Granick, 2009] and Wikipedia). The terms asymmetric and elliptical indicate that the given equations describe observations that are distributed differently in the two main directions where the peak is formed. For example, the distribution of the values in X direction is more narrow than the distribution of the values in Y direction.

<sup>4</sup> Perhaps there are other distributions more suitable for the modelling of the values in the score maps which the author is not aware of.

The parameters in the 2D elliptical equation are:

$A$  : Height of the peak

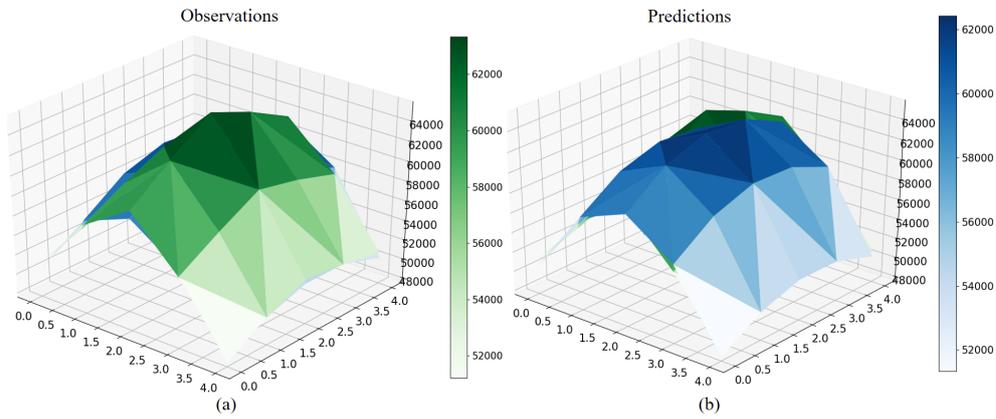
$(x_0, y_0)$  : Location of the peak

$a, b, c$  : Parameters that include the angle that describes the rotation of the peak, and the variance in  $X$  ( $\sigma_x$ ) and in  $Y$  ( $\sigma_y$ )

**DEVELOPED METHOD** The suitability of a Gaussian distribution is examined as explained below.

1. Firstly the score maps that passed the evaluation criteria are retrieved.
2. The peak pixel computed from previous methods is obtained.
3. The neighbouring region to the peak pixel is found. Some experiments were conducted to find out what is the ideal number of neighbouring pixel that must be taken into account. It was found that by using no more than 2 neighbours in each side of the peak pixel, the sub-pixel accuracy results were better than when higher number of neighbors had been used.
4. A least squares adjustment method is applied to find the optimal 2D elliptical Gaussian fit. The (bottom) equation in Figure 4.39 is used. An *elliptical* 2D Gaussian is used because the variance of the values in the two directions around the peak can be different in the score maps. An example is shown in the 'intensity gradient score map' of the Figure 4.37. To apply the least square adjustment, initial approximation of the unknowns are provided.  $A$  takes the value of the discrete highest peak value in the score map.  $(x_0, y_0)$  take the location of the discrete highest peak.  $a, b, c$  adopt random numbers. After some iterations, the sum of the squares of the residuals (between the observations and the predictions created from the fitting) is minimized. The input parameters that are initially approximate obtain final optimal estimations based on the least squares adjustment. The new  $(x_0, y_0)$  constitutes the location of the highest point in the fitted Gaussian surface. This is the matching location of the images from which the score map was retrieved. The location has sub-pixel accuracy.
5. The new matching location is converted to space coordinates as explained in Section 4.6.

The Figure 4.40 shows a visual example of how a 2D elliptical Gaussian is fitted on the similarity values around the peak value of a score map.



**Figure 4.40:** a) The observations from a score map are triangulated for visualization purposes. However, observations exist only at the vertices of the triangles. The observations include the discrete highest similarity value of a score map and two neighboring pixels at every side of the highest similarity. The score map used in the specific example was extracted from images with resolution 0.5m. The bar next to the graph indicates the corresponding similarity score for each gradation of green. The blue part that appears at the left, belongs to the fitted Gaussian surface, which is clearly illustrated in (b). b) The predicted values from the fitting of an elliptical 2D Gaussian. The predicted values are triangulated for visualization purposes. By observing the triangles, the differences between the observations shown in (a) and the predictions shown in (b) can be spotted. The bar next to the graph indicates the corresponding similarity score for each gradation of blue.



# 5

## IMPLEMENTATION & EXPERIMENTS

This chapter includes details about the implementation of the proposed image-based method for point cloud registration, which has been analyzed in Chapter 4. To determine whether the implemented method is suitable for the registration of 3D point cloud tiles, several experiments have been executed. Specifically, the chapter is split in three sections. In Section 5.1, the data-sets and tools used to perform the experiments are presented. In Section 5.2, specific details of the implemented prototype are explained. Finally, in Section 5.3, the conducted experiments are explained and the way the metrics of performance alternate is studied.

### 5.1 DATA-SETS USED & TOOLS

#### 5.1.1 Data-sets

This section provides information about the data-sets and the tools used in order to develop and test the proposed method.

Point cloud data from the West Paris in France and Schiedam in The Netherlands has been used for the experiments. The data, which is collected with a HDL-32E Velodyne scanner, is provided by the company CycloMedia Technology B.V. To allow the developed algorithm to deal with different cases, the point cloud tiles used for the experiments represent scans from different areas. Bridges with roads below, roundabouts, big crossings and narrow streets are some examples of the included cases.

The input to the algorithm is point cloud tiles each stored in a **LAZ format**. The LAZ files are converted to **LAS format** files. In total, 62 LAS files of square point cloud tiles that have size  $250m^2$  have been processed. Based on the number of overlapping LAS files, the number of point cloud pairs that have to be matched are computed. The number of point clouds pairs is 218. This, multiplied by 27 types of images gives a total of 5886 image pair registrations (Table 5.1).

#### 5.1.2 Software & Hardware

The proposed method was implemented in Python 2.7. Software libraries that were used directly and software packages are explained below.

**LASTOOLS** The software suite LAStools contains command line tools for point cloud processing. Specifically the tool 'laszip' was used to convert the **LAZ format** files to **LAS format** files. This step was necessary for the processing of the point clouds.

**SUBPROCESS** The subprocess library was used in order to apply the laszip tool (see previous paragraph) in the command prompt through Python.

Scene	LAS files	Point cloud pairs	Image matchings
1	10	45	1215
2	8	28	756
3	6	15	405
4	11	55	1485
5	3	3	81
6	6	15	405
7	4	6	162
8	10	45	1215
9	4	6	162
<b>TOTAL</b>	<b>62</b>	<b>218</b>	<b>5886</b>

**Table 5.1:** The table illustrates information about the data-sets used. The first column is a count for the scenes of overlapping point cloud tiles. Each scene refers to a square area of  $250m^2$ . The number of existing overlapping point cloud tiles for each scene is shown in the second column. The number of the constructed point cloud pairs for each scene is shown in the third column. For the calculation of the number of point cloud pairs in each scene see Section 4.3.3. The number of the image pairs constructed for each scene is shown in the fourth column. This is the result of the number of point cloud pairs per scene multiplied by 27, which is the number of image types created per point cloud.

**LASPY** The Laspy library was used to read the points' coordinates from the LAS files and also other attributes of the points. It was also used to create new LAS files. This is necessary, for example when the outliers are rejected and when the estimated translation parameters are applied to match the point clouds.

**NUMPY** The NumPy library is used to construct the 2D grids, which are required for the images creation. More information is found in Section 5.2.3.

**SCIKIT-LEARN** The scikit-learn library was used to compute the k-Nearest Neighbors<sup>1</sup>. A [kd-tree](#) was selected for that purpose. It was faster than other options (such as a ball-tree) provided by the scikit-learn library. After the nearest neighbors detection, an implementation of the Local Outlier Factor (LOF) by the method scikit-learn was used to reject outliers.

**SCIPY, STATS** The SciPy library and its Statistical function *binned statistic* was used for the binning of points in 2D grid cell and the calculation of a single value per grid cell.

**OPENCV** The library OpenCV was used to apply many image processing techniques needed such as the computation of images' gradient and the Gaussian smoothing. It was also used to apply the template matching method.

**SCIKIT-IMAGE** The scikit-image library was also used to process images.

**QUICK TERRAIN READER** The software Quick Terrain Reader was used to observe the point cloud tiles.

No multi-processing or any other library or technique that could minimize the execution time of the algorithm was used.

**HARDWARE** For the execution of the experiments, a computer with Core(TM) i7 processor, CPU power 2.7GHz and 16GB RAM was used.

<sup>1</sup> The [k-NN](#) are found and not the Fixed Distance Neighbors as this is a prerequisite of the [LOF](#) method (Section 2.4.1). The Fixed Distance Neighbors are the points that can be found within a fixed distance from the examined point.

## 5.2 IMPLEMENTED PROTOTYPE

All the implementation details of the proposed method are explained in the following subsections.

### 5.2.1 Input data

The input data was located in one main folder, assuming that the user of the algorithm will store the data in a single main directory. The main folder was consisted of sub-folders, representing the scenes. Each sub-folder was consisted of the overlapping point cloud tiles. This design decision was taken as it makes the iteration over different scenes easy.

After the outlier rejection, the inlier points are written in a new las file. The header of the input LAS file is used to create the new LAS file with the inlier points. The header defines which attributes of the points can be specified. Apart from the X,Y and Z coordinates of the inlier points, their intensity values is also written in the new file. Those new files are used for further process. The intensity data is kept because it is one of the points' attributes used to assign values in the created images.

### 5.2.2 Normal vectors

Following the computation of the normal vectors on the inlier points, the RGB component of the LAS files is used to store the normal vectors values. Prior to the storing, the normal values are normalized. The normal vector values, which range from -1 to 1, are normalized from 0 to 65535. The number 65535 is used because it is the maximum number that can be represented by the 16-bit uint16 (unsigned integer) type of OpenCV, which is the library used to write images. Following that, the normalized normal vector values are stored in the RGB component of the LAS files.

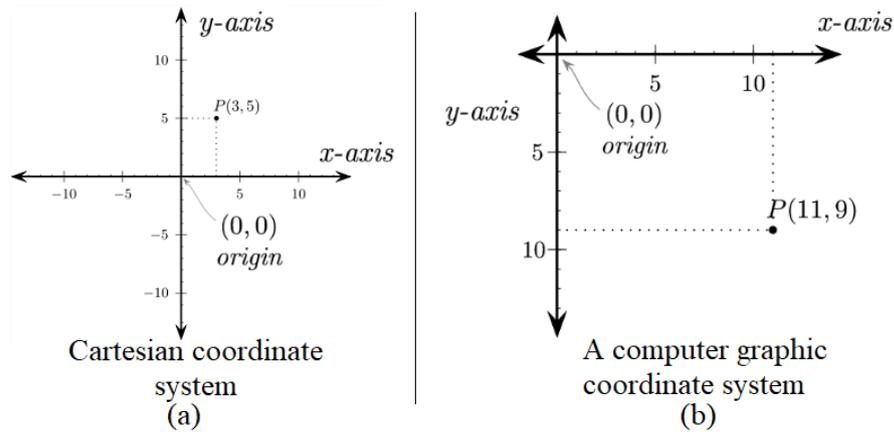
The normal vectors are described by three values indicating the size and orientation of the vectors in each one of the three main directions (X, Y, Z). The values in X directions are saved in the red component, the values in Y directions are saved in the green component and the values in the Z directions are saved in the blue component.

### 5.2.3 Reduction of dimensions

For the creation of 2D projections, arrays are created where each cell represents a pixel of an image. As mentioned, the library NumPy is used for the arrays creation. In Numpy, to assign values to each array cell, one should specify first the row index and then the column index of each pixel. The two coordinates of the points used to construct a specific 2D projection, determine their image coordinates. The coordinates of the y-axis of a projection represent the indices of the rows in a table. The coordinates of the x-axis of a projection represent the indices of the columns in a table. Therefore, the coordinates that required to be in the Y axis are specified before the coordinates that required to be in the X axis. In comparison, it is common practise to refer first to the X values and then the Y values. For example, for the construction of a YZ-projection the Z values must be specified first and then the Y values.

In order to create computer graphics or simply digital images it is necessary to define how the computer will handle X and Y coordinates. OpenCV library uses a computer graphic coordinate system which slightly differs from the well-known Cartesian Coordinate system (Figure 5.1). For this computer graphic coordinate system the X-coordinates are applied the same way as in the Cartesian. The X-coordinates increase from the origin of the system towards the right direction.

However, the zero Y-coordinate in the specific computer coordinate system lays at the top (of the computer screen). Simply put, as the Y values increase the computer graphic coordinate position moves down (the screen) [Craven, 2016]. In contrast, in the Cartesian coordinate system Y values increase as the coordinate position moves up.



**Figure 5.1:** a) The Cartesian Coordinate system. b) A computer graphic coordinate system (which is used by the library OpenCV). (Figures from [Craven, 2016]).

The height of the created images is represented by the Y-axis. The created projections must show the coordinates of the Y-axis as they are; the coordinates in the Y-axis must be increased from the bottom to the top. However, as the system used for the creation of computer graphics requires the low coordinates of the Y-axis to be shown on the top and the high coordinates of the Y-axis to be shown on the bottom, the 2D projections would be upside-down with respect to the Y axis. In order to avoid this, the coordinates that must be placed on the Y-axis of the projection get a negative sign. For instance, for the creation of an XY-projection, the Y coordinates of the points become negative. Therefore, during the binning process of the points into grid cells, a point representing part of the top of a building, instead of being binned in a grid cell on the bottom of the projection, is binned in a grid cell on the top of the projection.

#### 5.2.4 Image registration

The image registration is performed with a template matching technique. The template matching is applied through the OpenCV software library. Another restriction by the library is that the images that are compared must be 8-bit or 32-bit floating-point. As mentioned, the images are stored with unsigned integer 16-bit data type. In order to maintain the resolution, the data type of the images is converted to 32-bit floating-point. Then the template matching is applied and the retrieved similarity score maps are stored again with data type unsigned integer 16-bit. However, similarity values can be higher than 65535, which is the highest possible value with the data type uint16. The normalization of the values in the score maps up to 65535 may lead to false conclusions. Particularly, when the two highest similarity values of a score map are compared to determine whether their difference is significant (see page 68), but their difference has been shrunk due to the normalization. A relevant solution is suggested in Chapter 7.

## 5.3 EXPERIMENTS

This section contains information about the experiments executed to evaluate the performance of the proposed method.

The experiments are performed with absolute LOF threshold equal to 1.3. In other words, the 3D points of which the Local Outlier Factor score is lower than 1.3, are considered inliers. According to the literature, the closer the LOF score to 1, the higher the chance the point constitutes an inlier. However, experiments have shown that a threshold equal to 1 was excluding large number of points. Experiments with respect to the ideal LOF threshold is something that is considered for future work (Section 7).

Experiments were performed to examine different grid cell sizes of the created 2D projections. Particularly, the aim was to find out if any specific grid cell size produces better registration results than other grid cell sizes. It is expected that smaller grid cell sizes will give results of better quality because the smaller the grid cell size, the higher the resolution of the images, and the closer the 2D representation is to inputted the 3D point cloud tile. However, these are speculations and that's why the aforementioned experiments are performed.

To determine which grid cell size is more suitable for the image-base registration of point clouds, three metrics of performance are used. These are the execution time, the overall internal reliability (or robustness) of each estimated translation estimation and its standard deviation. Specifically, grid cells of size 0.05m, 0.15m and 0.5m are used.

**EXECUTION TIME - OVERVIEW:** In Figure 5.2, a comparison of the time needed to execute the main steps of the algorithm is presented. Approximately 101, 140 and 310 minutes required for the creation and registration of 5886 image pairs with image grid cell size of 0.5m, 0.15m and 0.05m correspondingly. The difference between the execution times of the implemented prototype with each grid cell size is significant.

**EXECUTION TIME - GRAPH EXPLANATION:** Each step that indicates an image type includes the reading of the 3D points, their binning into grid cells and the creation of the images. The depth images include also the creation of the depth gradient images. The intensity images include the retrieval of the intensity values from the LAS files and the generation of the intensity gradient images. The normal vector images does not include the computation of the normal vector. The normal computation is performed per point and therefore it takes extremely long time. Suggestions to improve this are mentioned in Section 7.2.

**EXECUTION TIME - GRAPH OBSERVATIONS:** The registration of images with cell size 0.05m needs almost double the time which is needed to register images with grid cell 0.15m. With all the grid cell sizes, the creation of the density images requires more time than the other image types. The time needed to create the density images with grid cell size 0.05m is more than two times the duration of the density images creation with grid cell size 0.15m. The detection of the optimal solution is almost constant for all the cells sizes as it is not depended on the amount of cells, but on the amount of the accepted image matchings.

**EXECUTION TIME - EXPLANATION OF OBSERVATIONS:** *Why are the total execution times of each experiment not proportional to the grid cell size? The same question is valid for the time need to construct each image type.* As mentioned in Chapter 4.5 it is common that convolution filters are not applied in the spatial domain. That means that when a convolution filter has to be applied on an image then the process followed is not the shifting of a window of pixel over an image. This process is extremely slow and therefore it is replaced by a convolution of the values in the

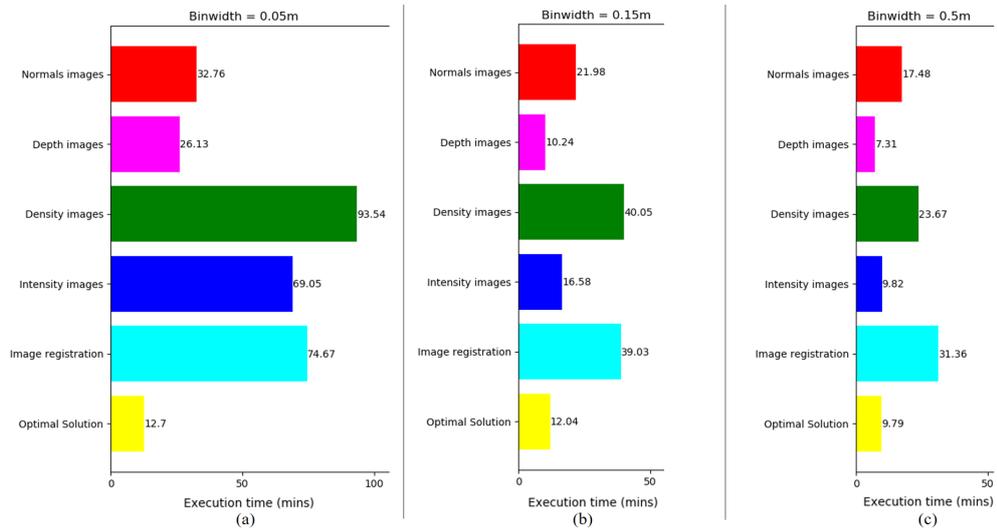


Figure 5.2: Execution time of algorithm's main steps when using grid cells size of 0.05m, 0.15m and 0.05m.

Frequency Domain via the Fast Fourier Transform. Due to this, the execution times are not solely related to the grid cell size. OpenCV which is used in this project employs the convolution in the frequency domain in its convolution filters used for the development of the proposed method. These are the Sobel Operator (image gradient), the Gaussian smoothing and the template matching.

*Why is the execution time needed for the construction of density images with resolution 0.05m significantly higher than that of other images<sup>2</sup>?*

The graph in Figure 5.2 was not created in a way to allow to answer this question with certainty. This is because, as mentioned, each step shown in Figure 5.2 includes multiple small steps. Thus, it is not clear which part of each generic step causes high or low execution times. However, an assumption that can be made is that the high execution time of the density images is caused by the counting of the points in order to generate one values per pixel. This is the only step which was only implemented for the density images and for not any other image. All the other sub-steps towards the creation of the density images were used also for the other images.

**QUALITY: PRECISION & INTERNAL RELIABILITY** To judge the suitability of different grid cell sizes (also mentioned as bin-widths), graphs that combine internal reliability with precision are constructed. As explained, the overall internal reliability is simply described with the amount of image types of which the matching result has passed the evaluation criteria (see Section 4.7, page 4.7). All the image types are 9 and the internal reliability is described as a fraction. If for example all the image types passed the evaluation criteria and thus all have been considered for the computation of the final estimation, then the overall internal reliability would be 9/9. The internal reliability is given on the x axis of the graphs in Figures 5.3, 5.4 and 5.5. The precision is given on the Y axis. The precision is described with the standard deviation of the included estimations for the computation of the optimum one.

In Figure 5.3 it can be seen that when the largest grid cell is used as shown in 5.3c, the internal reliability reaches its highest level which is '1', only once. In

<sup>2</sup> The time duration required for the execution of the step 'intensity images' is also high but this step includes both the construction of the intensity images and intensity gradient images.

comparison, the other two graphs which refer to results from grid cell size 0.05m (5.3a) and 0.15m (5.3b) show many values concentrated at the bottom right sides of the graphs. The values that lay at the bottom left sides refer to final translation parameters that have high accuracy and high internal reliability. Nevertheless, in Figure 5.3b which describes the quality of the estimations computed with grid cell size 0.15, there are more values that have poor (high) standard deviation.

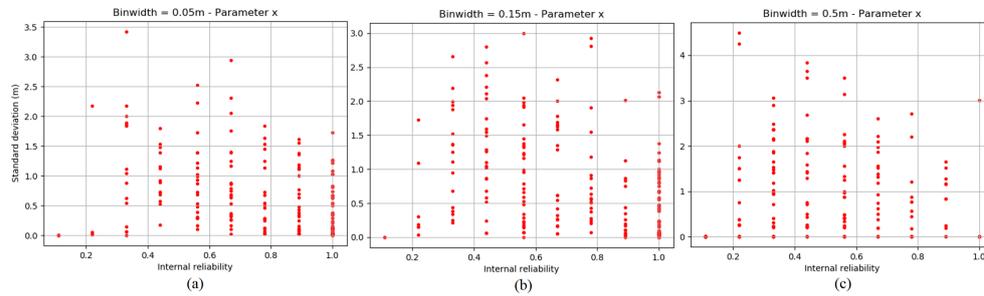


Figure 5.3: A scatter plot that shows the relationship between the internal reliability and the standard deviation of the X translation parameters computed with a) grid cell size 0.05m, b) grid cell size 0.15m and c) grid cell size 0.5m.

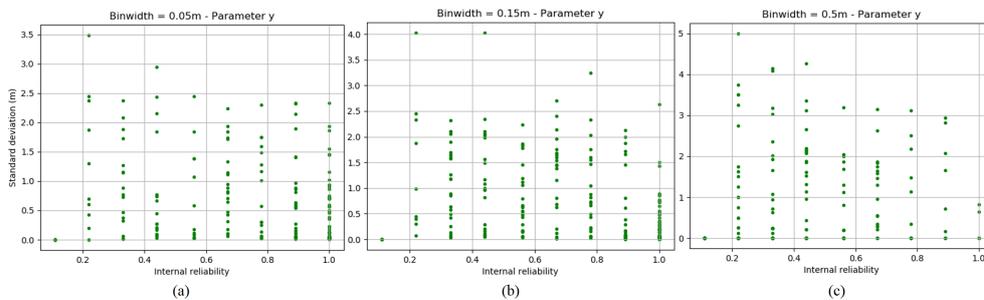


Figure 5.4: A scatter plot that shows the relationship between the internal reliability and the standard deviation of the Y translation parameters computed with a) grid cell size 0.05m, b) grid cell size 0.15m and c) grid cell size 0.5m.

The Figures 5.4 and 5.5 show similar patterns as the Figure 5.3. Especially in the Figure 5.5a, the values are more dense at the bottom of the graph, while in the Figure 5.5b the values are more scattered. This indicates that in 5.5a there are many values of which the internal reliability and the precision are higher than the values computed with 0.15m grid cell size.

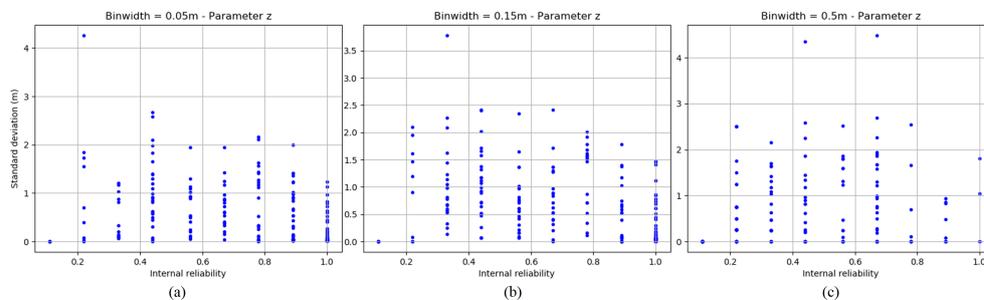


Figure 5.5: A scatter plot that shows the relationship between the internal reliability and the standard deviation of the Z translation parameters computed with a) grid cell size 0.05m, b) grid cell size 0.15m and c) grid cell size 0.5m.

The aforementioned experiments were performed to determine the most suitable grid cell size to create the 2D projections. As observed, the two quality measures are quite low when the larger grid cell size is used (0.5m). In comparison, the quality measures are higher when the grid cell size is 0.15m, and they are even higher when the grid cell size is 0.05m. However, the smallest grid cell size (0.05m) requires more than double the time needed to execute the algorithm with grid cell size 0.15m. What is best to choose it is always related to the application and to the user requirements.

# 6

## RESULTS & ANALYSIS

This chapter provides the results of experiments and analyses them. To judge the results of the proposed method, the smaller grid cell size (0.05m) is used in order to show the highest quality which can be retrieved with the developed method.

### 6.1 MAIN METHOD RESULTS

This section provides results from point cloud pairs matched with the proposed image-based multi-registration method. Different cases are considered. The first case is about point clouds with big overlap, but are different in the overlapping part due to the movement of cars. The second case is about point clouds with very big overlap, but with large offset. The third case represents a scene with a bridge, a parallel road and a perpendicular road. The fourth case shows point clouds with large overlap, large offset but distinguishable features like walls and light poles. The fifth case shows a point cloud retrieved in a tunnel and a neighbouring point cloud retrieved outside the tunnel.

#### 6.1.1 Case 1: Big overlap, changes in the scene, uncrowded scene

The first case is about two point clouds which have relatively big overlap. This can be seen by observing the two point clouds individually in Figure 6.1. In Figure 6.2a (before registration) it can be seen that there are many objects that appear twice. This indicates that there is offset between the point clouds. Figure 6.2b shows the result after the registration. It can be seen that the objects do not appear twice anymore. This indicates visually that the proposed method performed successfully, as the corresponding points are matched; after registration the corresponding objects appear only once when the two point clouds are superimposed. The Figure 6.3 shows the same conclusion, but arithmetically.

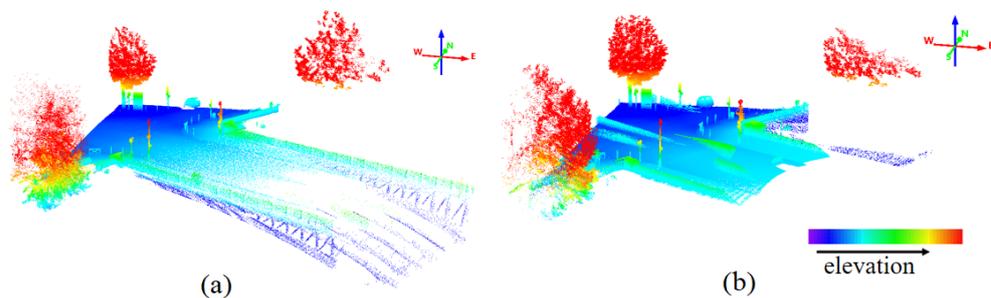
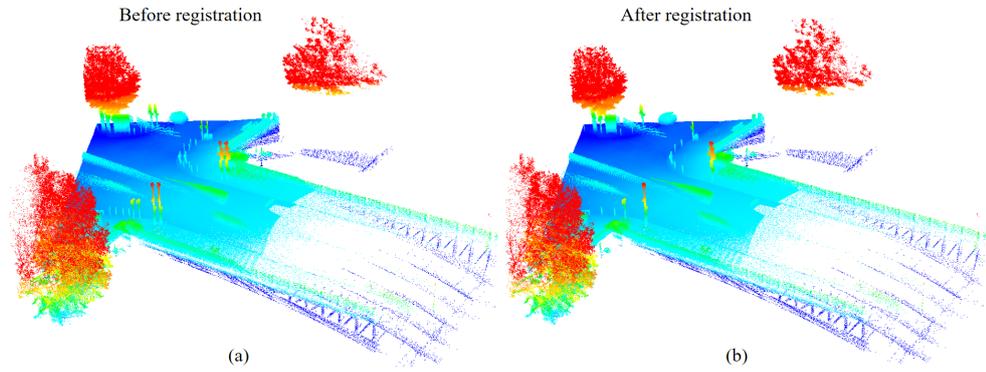


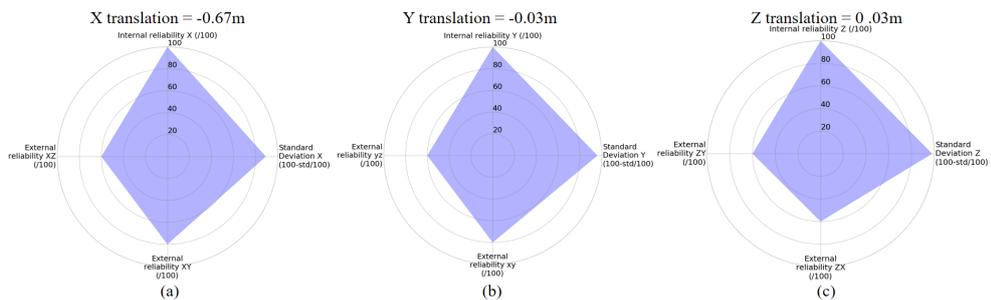
Figure 6.1: (a) Square point cloud tile of  $250m^2$ . (b) A second point cloud tile that captures the same region as (a) recorded at a different time. It can be seen that both have points representing the upper left part of the square tile. This means that they overlap approximately at the one fourth area of the tile. The point cloud (b) has points representing cars. This can be seen by observing the light blue and light green points that appear on the ground which is illustrated with dark blue.

Specifically, the Figure 6.3 illustrates with clockwise orientation (starting from the top) the internal reliability, the precision, the external reliability for the XY



**Figure 6.2:** a) Two point cloud tiles before registration. The poles are shown in double, which is an indication of the offset between the point clouds. b) The two point cloud tiles of (a) after registration.

projection pairs and the external reliability for the XZ projection pairs. As already explained the overall internal reliability indicates the amount of the image types matchings that passed the evaluation criteria and thus were taken into account for the calculation of the final parameters. The bigger possible internal reliability measure is 9 out of 9 while the smaller is 0 out of 9. Estimations with internal reliabilities equal or higher than 6 out of 9 are empirically considered reliable solutions. For visualization purposes the internal reliability values are multiplied by 100. The standard deviation illustrates the variability of the estimations considered for the calculation of the final parameters. The smaller the standard deviation, the higher the precision. For visualization purposes the standard deviations are multiplied by 100 and the result is subtracted from 100. As a result, small standard deviations cover large areas of the chart. Next, the external reliabilities are included. The measures for the external reliabilities, are simply the percentage of the covered area in the images (see Section 4.7). Since two projections are used for the computation of each parameter, two external reliability values are included.



**Figure 6.3:** The charts show the four quality measures for the final X, Y and Z estimations for the matching of the point cloud pair shown in 6.2. The bigger the measure, the more extended the purple area and the better the quality of the results. a) The chart shows the quality measures for the X translation. On the top of the graph the x translation estimation is indicated. b) The chart shows the quality measures for the Y translation. On the top of the graph the Y translation estimation is indicated. c) The chart shows the quality measures for the Z translation. On the top of the graph the Z translation estimation is indicated.

### 6.1.2 Case 2: Big overlap, large offset, crowded scene

The example in this case refers to point clouds that have a big overlap, but they have large offset in the Z axis. Additionally, the scene is crowded, in a sense that there is an abundant amount of information from points. In other words, there are millions of points in these two tiles. Due to that, it is possible that distinct features, such as light poles, which could help the registration, are covered by other features. For example, in an XZ orthogonal projection, the objects included in the upper right black rectangle of Figure 6.4 would be hidden by the bushes (shown in the big rectangle at the right).

Figure 6.4a shows the two point cloud tiles before the registration and Figure 6.4b shows the two point cloud tiles after the registration. Visually the results seem to be successful as the difference between the corresponding objects seems to be reduced.

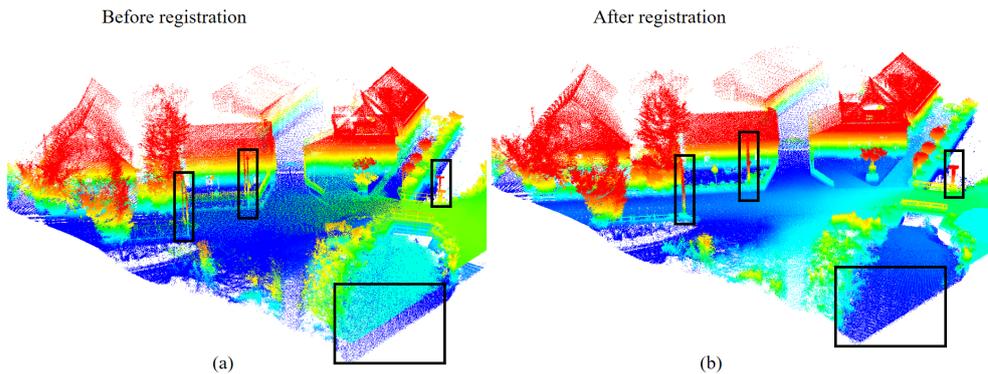


Figure 6.4: a) Two point cloud tiles before registration. The squares indicate the double information, or in other words the offsets between the two point clouds. b) The two point cloud tiles of (a) after registration.

The Figure 6.5 indicates the quality measures of the translation estimations of the point cloud pair shown in Figure 6.4.

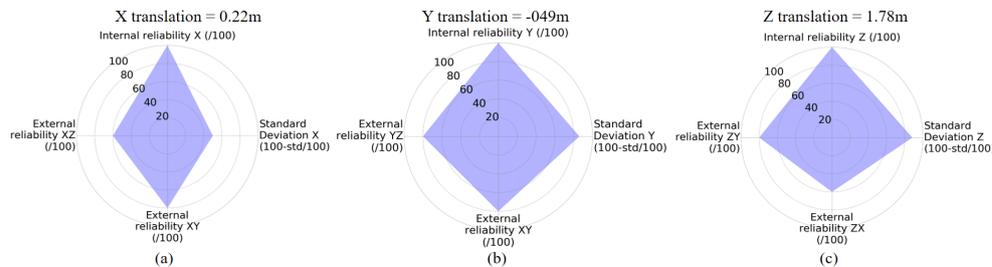
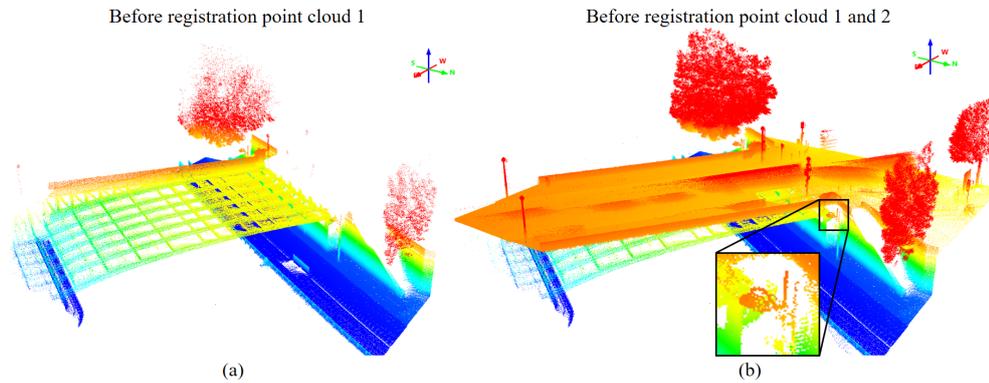


Figure 6.5: The charts show the four quality measures for the final X (a), Y (b) and Z (c) estimations for the matching of the point cloud pair shown in 6.4. The internal reliabilities of all the three parameters are equal to maximum possible value (9/9). However the standard deviations indicate that the final estimations have standard deviations not equal to zero. This means that the quality of the results is not at its top level, however there is an indication given with the standard deviations about the possible variance of the parameter from the truth.

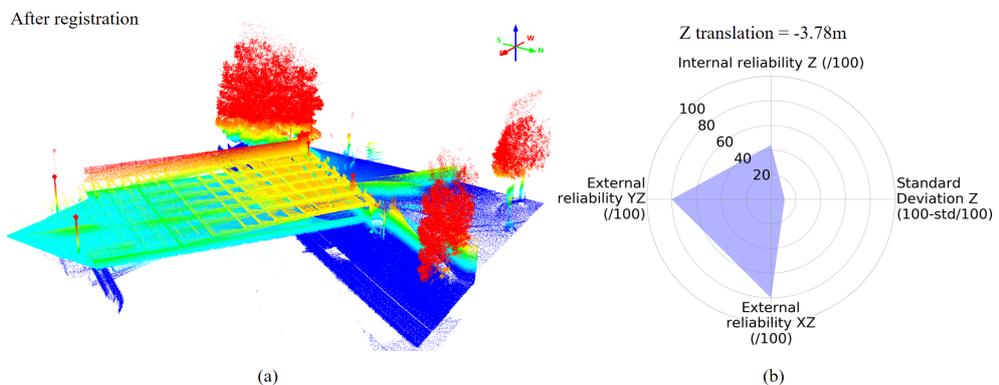
### 6.1.3 Case 3: Small overlap, presence of a bridge and a road

This case illustrates an example of a point cloud that represents a road and a bridge (Figure 6.6a) and a point cloud that represents a road on top of the bridge (Figure 6.6b). Furthermore, the overlap between the two point clouds as it can be observed in Figure 6.6b is significantly small compared to the tile area ( $250m^2$ ).



**Figure 6.6:** a) One point cloud tile. b) The point cloud tile of (a) overlaps with another point cloud. The two are illustrated here before registration. The magnified part stresses some overlap of the two point clouds and the small translation error between the two. The error is indicated by the double line of the pole.

After the registration of the point clouds shown in Figure 6.6, the results are produced and visualized in Figure 6.7a. The matching is completely unsuccessful since the road which was supposed to be on top, ended up below the bridge. The bad quality of the results is indicated with the quality measures in Figure 6.7b. Both the internal reliability and the precision have very low values. However the external reliability values are quite good, which is not representative of the results' quality. The method used to compute the external reliability measures, which indicates the influence of undetected errors, takes into account only one possible reason for undetected errors (see page 73). Since the problem here is not the amount of the captured points, but the very small overlap, the measures for the external reliability do not capture the truth about the quality of the estimations.



**Figure 6.7:** a) The point cloud tiles of the Figure 6.6 after the registration. b) The quality measures that describe the Z translation estimation. The Z is selected to be visualized because of the big failure in the matching in the Z axis.

#### 6.1.4 Case 4: Big overlap, large offset, distinct structures

The point cloud pair matched in this example is described by a big overlapping area between the two, but also a large offset. Moreover as it can be seen in Figure 6.8a there are many distinct structures that could facilitate the image matching. Specifically there are many poles and long walls. After the execution of the developed algorithm the result of the matching is given and it is depicted in Figure

6.8b. The visualization of the results indicates that the matching was successful. The same observation is made by looking at Figure 6.9.

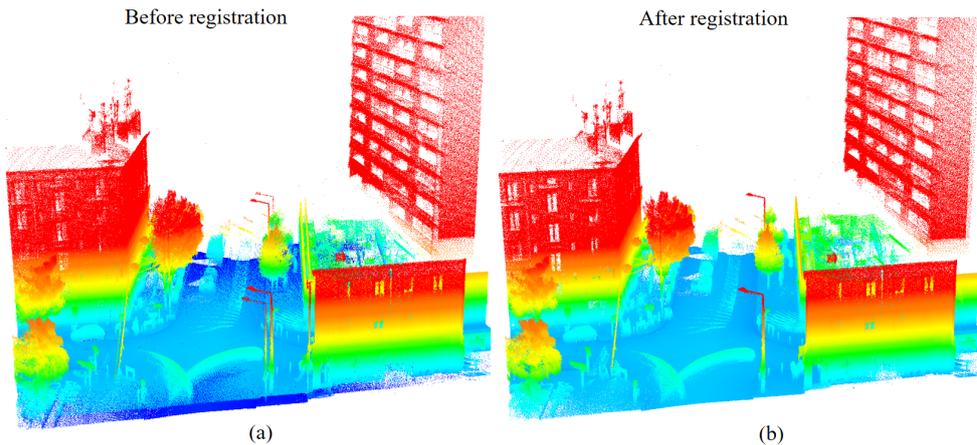


Figure 6.8: a) Two point cloud tiles before registration. b) The two point cloud tiles of (a) after registration.

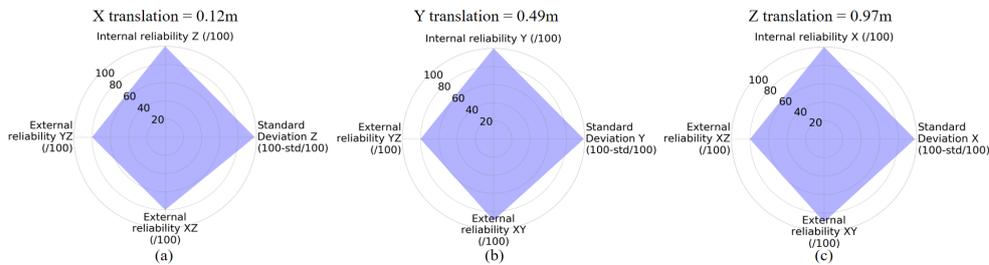


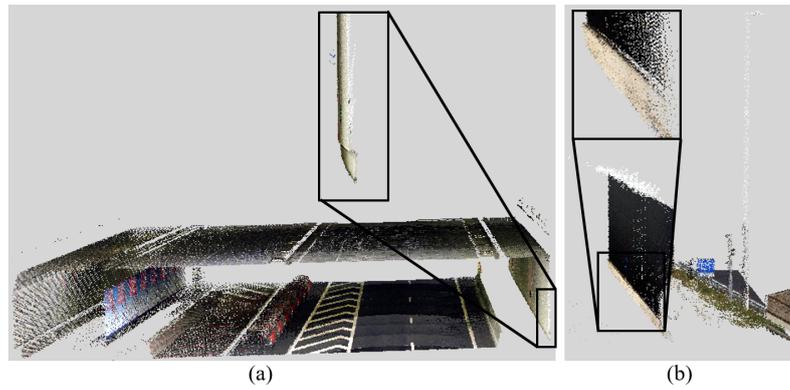
Figure 6.9: The charts show the four quality measures for the final X (a), Y (b) and Z (c) estimations for the matching of the point cloud pair shown in 6.8. Almost all the quality measures reach their maximum level. This means that the matching is arithmetically successful.

#### 6.1.5 Case 5: Absence of overlap, presence of tunnel

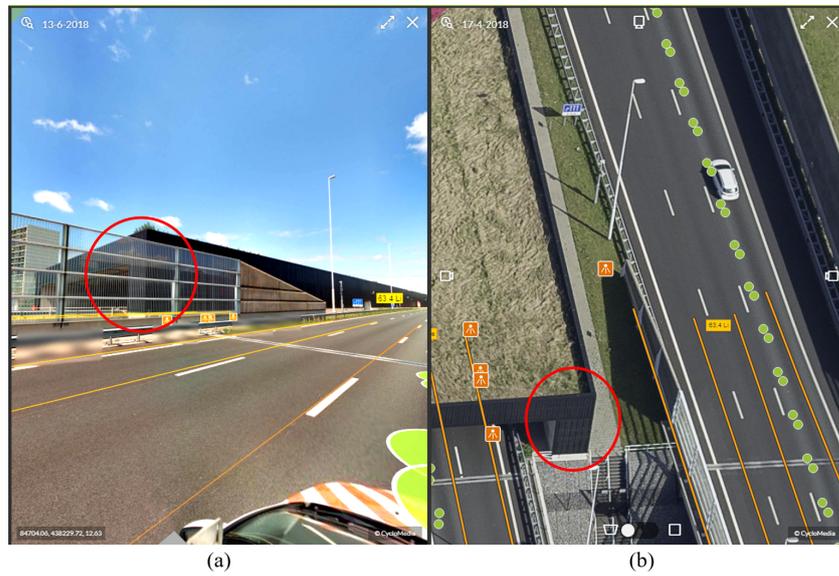
The example point cloud pair given here shares no overlap. However, the first impression when observing the two point clouds is that they share a wall. However, when observing the two walls carefully, it seems that the two walls do not represent the same wall (Figure 6.10).

To validate the assumption that the two point clouds do not share the same wall, a panoramic and an aerial image of the scene are observed (Figure 6.11). By doing so, it becomes clear that the two point clouds do not share any overlap, but they were considered as an overlapping point cloud pair because they are neighbors.

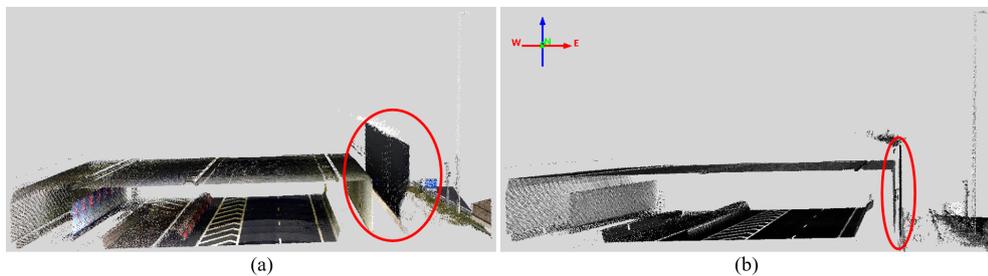
Figure 6.12a shows the two point clouds before the registration and Figure 6.12b shows the two point clouds after the registration. Before the registration, the distance between the two walls is visually comparable to the thickness of the tunnel's wall illustrated in Figure 6.11. In contrast, after the registration, the two walls are located close to each other but they are not match. The algorithm attempted to match the two walls as it is the only common information between the two point clouds, but it did not completely match as the structure of two walls differs at the bottom (Figure 6.10).



**Figure 6.10:** a) One point cloud tile scanned inside a tunnel. b) A neighbouring point cloud captured from a road next to the tunnel. A first impression is that the wall on the right side of point cloud (a) is the same wall as the wall on the left side of point cloud (b). However, when observing the magnified parts of each wall it can be seen that they differ at their lower part.

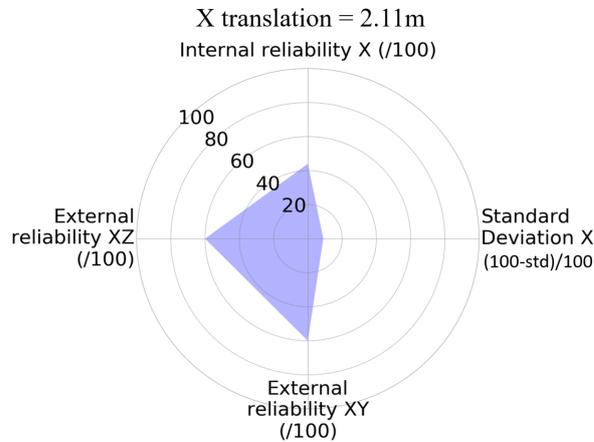


**Figure 6.11:** a) A street view image showing a tunnel within which the point cloud 6.10a was recorded. It also shows the information captured by point cloud 6.10b, as we can see the light pole and the blue sign. b) In the aerial image the tunnel wall is slightly more distinct. The red circles include the area of interest. It can be seen that the wall captured in point cloud 6.10a represents a wall inside the tunnel, while the wall captured in point cloud 6.10b represents a wall outside the tunnel. (Images from StreetSmart online viewer, CycloMedia 2018.)



**Figure 6.12:** a) The two point cloud tiles of Figure 6.10 before the registration. b) The two point cloud tiles of Figure 6.10 after the registration.

The registration result produced for this case with the developed algorithm would be successful if the quality measures of the two point clouds are poor. This statement is made because the visual result shows that the point clouds are shifted closer to each, but as explained this is not the real case. The Figure 6.13 is observed. The two main quality measures indicate that the registration result cannot be trusted, which is indeed the case.



**Figure 6.13:** The chart shows the four quality measures for the final X translation estimation of the point cloud pair shown in Figure 6.10. The focus is on the X direction because the two walls captured in the two point clouds are perpendicular to the X axis. The standard deviation points at 10 which means that the real value is 0.9m  $((100 - 10))/100$ . An estimation with precision 0.9m is not considered precise at all, as the point clouds accuracy is 2cm. The internal reliability of the X translation is 44 out of 100, which means that from the 9 type of images, the 4 have passed the evaluation criteria. This indicates a medium internal reliability. In combination with the standard deviation however, the conclusion is that the specific result cannot be trusted.

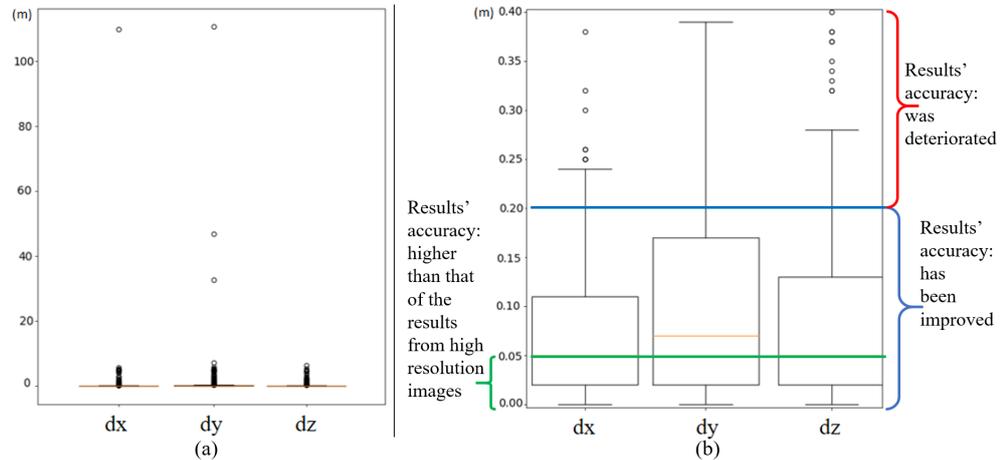
## 6.2 SUB-PIXEL ACCURACY RESULTS

This subsection contains information about the experiment performed to judge the quality of the sub-pixel accuracy method. Specifically, it is examined to which extent the transformation parameters resulting:

- from registering imagery of low resolution and applying the proposed sub-pixel accuracy method, could approach the transformation parameters resulting
- from registering imagery of high resolution.

The grid cell size of the low resolution imagery used for the experiments was 0.2m, and 0.05m for the high resolution imagery. Figure 6.14a indicates with box plots the absolute differences in X, Y and Z between the two approaches. In other words, the difference of each parameter computed from each of the two methods is calculated. Four large outliers are encountered. For visualization purposes the Figure 6.14a is magnified as shown in Figure 6.14b. The sub-pixel approach is completely successful for dx, dy and dz that are equal or lower than 0.05m. That is because 0.05m is the accuracy which can be retrieved with the high resolution imagery (i.e. 0.05m grid cell size). More than 25% of the dx, dy and dz are equal to or less than 0.05m. These are included in the set coloured with green. The sub-pixel approach has improved the accuracy of the parameters for which the dx, dy and dz are lower than 0.2m, because this is the accuracy which could be retrieved with the low resolution imagery (i.e.

0.2m grid cell size). More than 50% of the dx, dy and dz are smaller than 0.2m. These are included in the set coloured with blue. This indicates that the accuracy of more than 50% of the translation parameters was enhanced. Nevertheless, 20cm accuracy is considered low as the accuracy of the point clouds is 2cm. Finally, a large amount of translation parameters computed with the sub-pixel accuracy method and low resolution images, deviate significantly from the parameters computed with high resolution images. These are all the parameters that have dx, dy, dz higher than 0.2m which is the accuracy of the high resolution images used in the experiment. These are included in the set coloured with red.



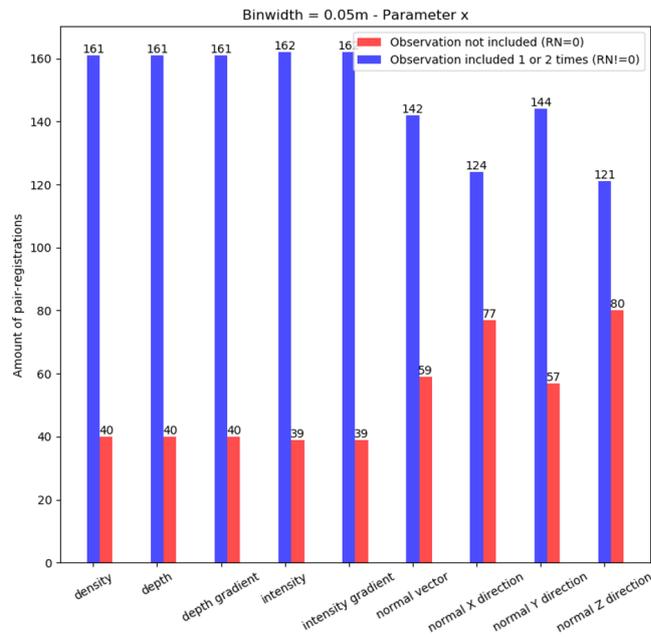
**Figure 6.14:** Differences between 1) the translations resulted from the registration and the sub-pixel accuracy method of images with low resolution and, 2) the translations resulted from the registration of images with high resolution 0.05m

### 6.3 IMAGE-TYPE SUITABILITY FOR REGISTRATION

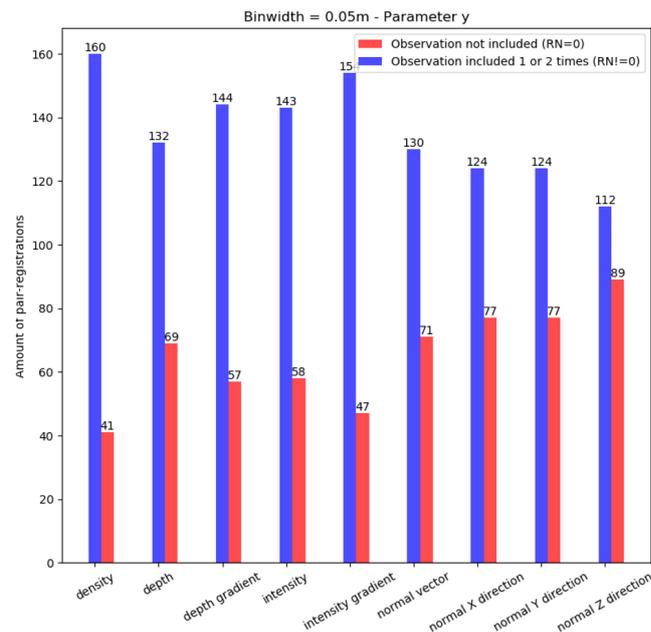
The proposed method employs a multi-registration method. Several image types are used to ultimately match point clouds. The results are analyzed to find out whether specific image types are more suitable for the registration of mobile scanned point clouds. For this purpose, the internal reliabilities calculated per image type are used. The internal reliability of each image type equals the redundancy number of the estimated parameters per image type. Simply put, the redundancy number equals the amount of estimations from a specific image-type that are taken into account for the calculation of the optimum solution. For more information see page 71.

The Figure 6.15 illustrates the redundancy number of the estimated X translation errors. It can be seen that for the X translation parameters the amount of registrations that were completely rejected were almost equal for the density, depth, depth gradient, intensity and intensity gradient images. In contrast, the matchings of the normal vector images had to be rejected a lot more times. This indicates that the normal vector images are less suitable for image registration at least for the projections that include the X coordinates. These are the XY and XZ projections.

The Figure 6.16 illustrates the redundancy number of the estimated Y translation errors. It can be seen that in comparison with the X translations, the estimations for the Y translation were rejected more times. Again the normal vector images are those from which the estimations are rejected more often. Nevertheless, the estimations from the normal vector images are rejected a few less times than the estimations from the images that show the normal vector values in the 3 directions. Also the estimations from the depth gradient images are rejected less times than the estimations from the depth images. The same observation is valid for the estimation



**Figure 6.15:** Redundancy number of the estimated X translation errors per image type. The x axis of the graph indicates the image type. The y axis of the graph shows the amount of registrations. The blue bars represent the amount of registrations from which both or at least one of the XY and XZ-projections' matchings, passed the evaluation criteria. The red bars represent the amount of registrations where none of the XY and XZ-projections' matchings passed the evaluation criteria.

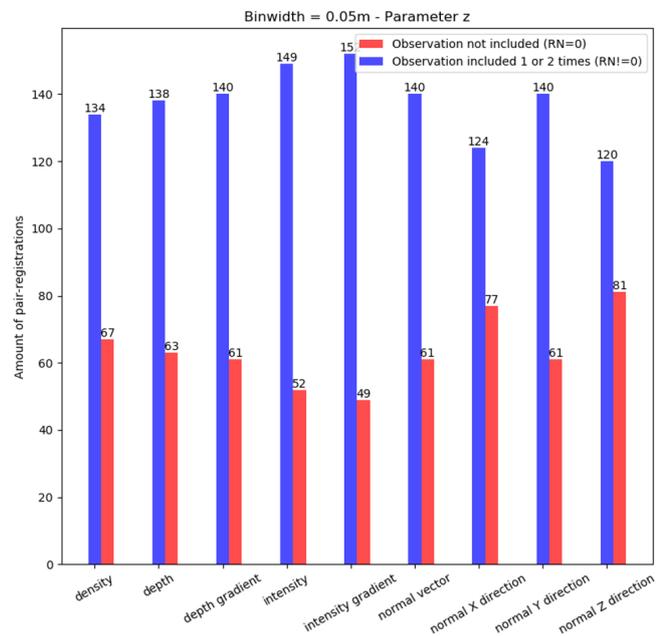


**Figure 6.16:** Redundancy number of the estimated Y translation errors per image type. The x axis of the graph indicates the image type. The y axis of the graph shows the amount of registrations. The blue bars represent the amount of registrations from which both or at least one of the XY and YZ-projections' matchings, passed the evaluation criteria. The red bars represent the amount of registrations where none of the XY and YZ-projections' matchings passed the evaluation criteria.

from the intensity gradient and the intensity images. This indicates that the gradient of the depth and the gradient of the intensity images provided better registration results than the depth and the intensity images. However, more

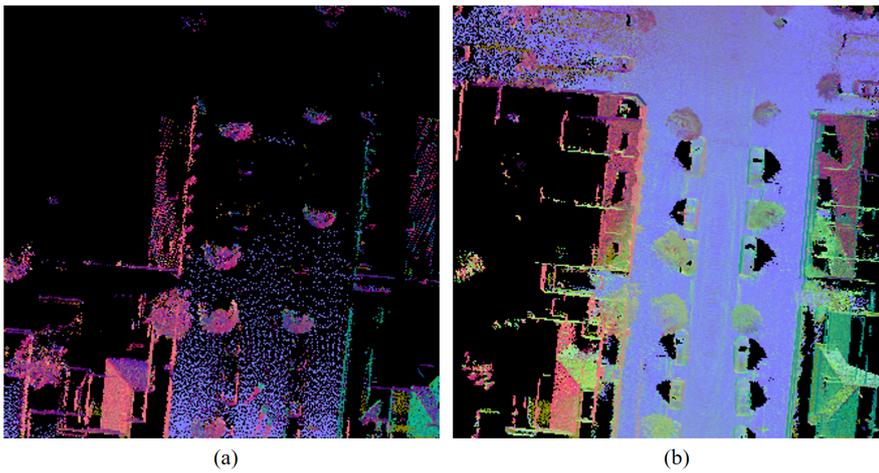
registrations from depth gradient images than intensity images were rejected. This indicates that the intensity attribute is better than the depth attribute of the 3D points. Finally, it can be seen that the  $y$  estimations computed from the density images have been accepted more times than the estimations from any other image type. This indicates that that the score maps of the density registrations had better quality. In other words, the highest similarity value between the two matched images were more distinct. Therefore the density images were more suitable than any other image type, at least for the calculation of the  $y$  translation parameter.

The Figure 6.17 illustrates the redundancy number of the estimated  $Z$  translation errors. Like it was previously observed, the gradient of the intensity and the gradient of the depth images, provided better results than the intensity and the depth images, respectively. Also the registration results from the intensity images were more times accepted than the results from the depth images. Again the estimated translations from the normal vector images were the most rejected. The estimations from the normal vector images are rejected a few less times than the estimations from the images that show the normal vector values in the 3 directions. The only difference for the  $Z$  translation than for the  $X$  and  $Y$  translations, is that many density ( $XZ$  and  $YZ$ ) projections were rejected.



**Figure 6.17:** Redundancy number of the estimated  $Z$  translation errors per image type. The  $x$  axis of the graph indicates the image type. The  $y$  axis of the graph shows the amount of registrations. Particularly, the blue bars represent the amount of registrations from which both or at least one of the  $XZ$  and  $YZ$ -projections' matchings, passed the evaluation criteria. The red bars represent the amount of registrations where none of the  $XZ$  and  $YZ$ -projections' matchings passed the evaluation criteria.

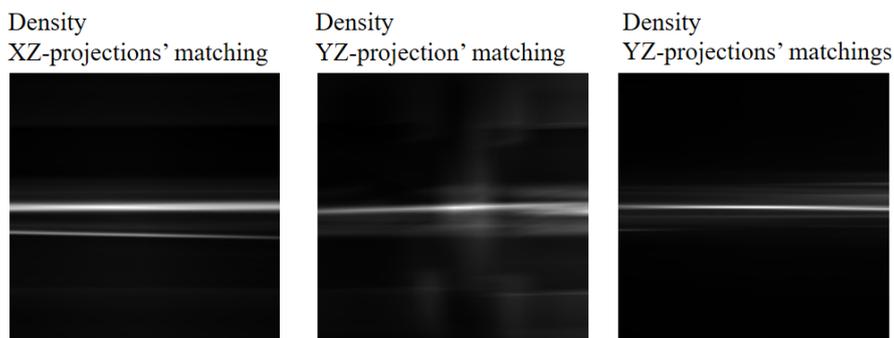
**WHY NORMAL VECTOR IMAGES FAIL MORE OFTEN THAN OTHER IMAGES?** The reason why the normal vector images fail to produce as good results as other images is due to the method used for their computation. The problem is the way the normal vectors are orientated. The normal vectors' orientation is based on the trajectory of the points. If two overlapping point clouds need to be matched and their trajectory is different, which is most of the time the case, then identical features could be illustrated with different normal vector values (Figure 6.18).



**Figure 6.18:** Two XY-projections that illustrate the normal vectors of the 3D points. The two projections originate from two overlapping point clouds retrieved from different trajectories. Therefore, the orientations of the normal vectors of corresponding points is different. This results to images where corresponding objects have different normal vector value. For example the trees are illustrated in the left image with pink, while in the right image are illustrated with green.

#### WHY DENSITY IMAGES FAIL WHEN XZ AND YZ MATCHINGS ARE PERFORMED?

The density projections are often rejected when comes to matchings that include the Z direction. This happens because only the Z direction is matched properly. To explain this, the Figure 6.19 is observed. The given score maps resulted from the matchings of XZ and YZ density projections. It can be seen that there is not a single peak to indicate the highest similarity value in the score maps. However, the matching location in the vertical direction is very distinct. In contrast, in the horizontal direction the registration method could not detect high similarity values. Since the evaluation criteria that test the quality of the score maps examine distinct peaks of values, the whole score map is rejected. That means that not only the X, but also the Z translation estimation is rejected.



**Figure 6.19:** Example of rejected score maps resulted from the registration of density XZ and density YZ projections. The specific score maps indicate that the Z translation parameter is sufficiently computed. This is shown by the high similarity values (illustrated with bright pixels) in the Z direction of the score maps. In contrast, in the Y direction of the images, there are no high similarity values.



# 7

## CONCLUSIONS & FUTURE WORK

This chapter presents the conclusions and the suggested future work of the conducted research. The conclusions are given in Section 7.1. This section provides the answers to the research questions of this thesis and summarizes the contributions of this work. The future work is presented in Section 7.2. It includes recommendations on the aspects that could be improved in the developed method. It also contains suggestions on how the specific project could be continued.

### 7.1 RESEARCH QUESTIONS

This section provides the conclusions of this project by answering the research questions.

1. *How to deal with the outlying points in the data so that they do not affect the image-based registration?*

The outlier rejection is tackled by using the Local Outlier Factor (LOF) method. This method computes the density of points in the neighbourhood of each point. To compute the LOF score of each point, the point's density in a neighbourhood is compared with the densities of the neighboring points. The closer the LOF score to 1 the higher the chance that the point is not very isolated, which means that the point is an inlier. The method used is suitable for rejecting noise from the point clouds, especially when the density of an outlier point is different from a neighboring point. For that reason the selection of the number of nearest neighbors is critical and varies between different applications. The method used is not suitable for the rejection of big outliers, such as moving objects. Moving objects can be cars, pedestrians, birds or even smoke.

2. *How to generate images from the point clouds and what kind of images so that the point clouds are best described?*

The developed research suggests the creation of images that project each point cloud tile on three planes; a XY-plane, a XZ-plane and a YZ-plane. The created projections show different attributes of the points. The attributes used are the density of the points, the intensity, the depth, the gradient of the intensity, the gradient of the depth and the calculated normal vectors of the points. Apart from the images based on the normal vector, the normal vectors are used to generate images that show the vector's value only in X, only in Y and only in Z direction. According to the results presented in Section 6.3, some image types perform better than others. The contribution of each image type to the registration of point cloud tiles is concluded below.

**A. DENSITY IMAGES** The registration results for the X and Y translations calculated from the density images, have passed the evaluation criteria more times than any other image type 6.3. In other words, in the score maps produced from the registration of the density XY-projections, the matching locations were described by the most distinct peaks and highest similarity values.

However, the estimated Z translations given from the density images were rejected based on the evaluation criteria more times than the results from other image types. The Z translations are calculated from the XZ and YZ projections. As shown in some example score maps in the results (Section 6.3), the matching of the XZ and YZ projections is successful in the Z direction but not in X and Y, correspondingly. The conclusion that can be drawn is that the density XZ and YZ projections are in many cases suitable for the calculation of the matching solution in Z but not for the matching in X and Y. Even though the Z translations may be calculated successfully, the method used to evaluate the results through the evaluation of the score maps, rejects both translation parameters. For example, if the density XZ-projections are matched and the resulted score map is rejected, then both the X and the Z translation estimations are rejected. However, there is a possibility that the Z is calculated correctly. Therefore, the evaluation criteria of the proposed method could be improved.

**B. INTENSITY AND INTENSITY GRADIENT IMAGES** The conclusion that can be drawn is that, at least for the Y and Z translation parameters, the gradient of the intensity images produce registration results of higher quality than the results from the intensity images. As shown and explained in Section 6.3 a higher number of intensity images, than intensity gradient images, have been rejected based on the evaluation criteria. Thus, if there was a necessity to choose one from the two, then that would be the intensity gradient image type. The reason why this is happening is because the gradient intensity images are less dependent on the intensity values and more dependent on the edges of the features constructed based on the intensity values. This is beneficial because, as explained in Section 4.4.4 and illustrated in Figure 4.15, the intensities of corresponding objects in two overlapping point clouds can be different due to the distance of the recording vehicle to the object.

**C. DEPTH AND DEPTH GRADIENT IMAGES** The conclusion that can be drawn for the depth-based images is that, at least for the Y and Z translation parameters, the gradient of the depth images produced registration results of higher quality than the results from the depth images. As shown and explained in Section 6.3 a higher number of depth images, than depth gradient images, have been rejected based on the evaluation criteria. Thus, if there was a necessity to choose one from the two, then that would be the depth gradient image type. The reason why this is happening is because the gradient depth images are less dependent on the depth values and more dependent on the edges of the features constructed based on the depth values. This is beneficial because, as explained in Section 4.4.6, the depth values of corresponding objects in two overlapping point clouds can be different due to the possible translation error between the coordinates of two overlapping point clouds. This is a disadvantage for the depth images because the depth is simply the third coordinate. For example, in a XY-depth projection, the Z coordinates represent the depth values.

**D. NORMAL VECTOR IMAGES AND IMAGES BASED ON THE THREE COMPONENTS OF THE NORMAL-VECTORS** These were the images from which the registration results were being rejected most of the times. This does not mean that the normal vector images were completely useless. However, their quality was sometimes lower than that of the other image types. Additionally, The images that have been constructed based on the three components of the normal-vectors, did not really facilitate the registration process. Most of the times, a lot more score maps are accepted when the basic normal vectors images are matched, than when the images based on the three components of the normal-vectors are matched. Consequently, it is concluded that the main normal vector images are suitable enough for the registration of images created from point clouds. The images that are based on the three components of the normal-vectors could have been neglected.

*3. How to compute the translation parameters that align relatively 3D overlapping point cloud pairs by matching 2D imagery?*

By applying a template matching method while using a simple correlation measure, the pixel where one image is placed on the other, can be found. The local registration of an image pair gives the translation estimations of the parameters used in the projections. If XY-projections are matched, then the matching location of X and Y is computed in pixel units. The pixel units are converted back to space units (meters) and then the translation parameters are retrieved. The proposed method suggests a multi-image registration technique, as multiple images are constructed from each point cloud pair. As a result, the X, Y and Z are computed many times. For that reason, a technique to detect the optimal solution for each parameter is applied. The technique assess each single image registration solution by applying criteria that evaluate the quality of the output. The criteria evaluate the strength (similarity value) and the uniqueness of the achieved matching. Only the results that pass the evaluation criteria are considered for the calculation of the optimum final solutions.

Although there are other commonly used methods for matching imagery, such as the extraction and matching of features, in this project a cross correlation template matching was used. Since the point clouds do not suffer from big rotation errors but small rotation errors might be possible (see Section 1.1.2), the template matching method had the capability to match point clouds. If point clouds with large rotation errors are used, then the simple template matching method wouldn't be able to perform successfully.

*4. How to assess reliability, precision and efficiency of the developed method automatically?*

The quality measures are computed which accompany each X, Y and Z translation estimation. As the method employs a multi-registration method, repeated and redundant estimations are produced. This facilitates the determination of the reliability and precision. The reliability of the method is assessed by providing metrics for the internal and external reliability of the optimum results. These two aspects of the reliability indicate correspondingly if an estimation is strongly checked and what is the influence of possible undetected errors on the final estimations.

The internal reliability equals the number of accepted estimations among the repeated estimations. This metric of quality performs successfully as it is correctly indicating if a final estimation is strongly checked (see Section 6.1). There is flexibility on what is considered bad or good internal reliability. For this project it was considered based on empirical observations that an internal reliability equal or higher 6 out of 9 observations could be highly trusted. However, this assumption was based on a few observed examples, therefore a more sophisticated method to determine the ideal threshold would be beneficial.

For the determination of the external reliability, causes of possible undetected errors are listed. Then, two external reliability weights are given to each image pair. The weights express the possibility that the translation estimation could be influenced by the number of points included in a tile. There are many other causes of possible undetected errors, for example the differences between the overlapping point clouds due to moving objects, but these are not currently taken into account. Therefore, as noticed also in some cases illustrated in Section 6.1, the external reliability weights do not always represent correctly the quality of the final estimations.

The precision is determined with the standard deviation of the estimations that pass the evaluation criteria. The smaller the standard deviation, the less the

variability between the considered values and the more precise the final registration estimation. Since the point clouds have accuracy of 2cm, a measure is considered highly precise if the standard deviation is equal to a few centimeters. However, this depends on the user requirements and use of the point clouds after their registration. In any case, the quality measures are computed and the user has the capability to deal with them as desired. This is valid also for the reliability measures.

To access the efficiency of the method, the execution time of the algorithm was tested when using different grid cell (pixel) sizes for the created imagery. Also the internal reliability and standard deviation measures were used to judge the overall quality of the developed method when using different grid cell sizes. It was noticed that the smaller the grid cell, the longer the duration of the execution time, and the higher the quality of the results. However, it was also noticed that the execution times were not proportional to the number of pixels contained in the images of different resolution as it was expected. As explained in Section 5.3 the reason why this is happening is because the convolution filters used in this project (such as the Sobel Operator for the image gradient, the Gaussian smoothing and the template matching) are implemented (by OpenCV software library) in the Frequency Domain via the Fast Fourier Transform. This method is faster than the convolution in the spatial domain where a window of pixels is shifted over an image to calculate new values. Additionally, the convolution in the Frequency Domain, as proven by the results (Figure 5.2), is less dependent on the number of pixels.

*5. The accuracy of the transformation estimations will be restricted to the grid cell size of the generated imagery. How to retrieve a sub-pixel accuracy results to improve the drawback of discrete grid cell size in the imagery?*

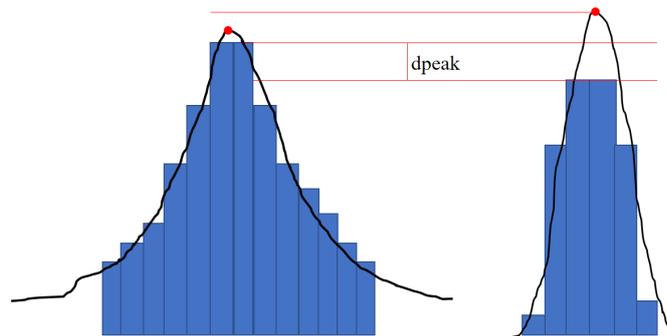
A 2D elliptical Gaussian function is used to model the similarity values in the neighbourhood of the highest similarity value. This constitutes the best matching location of one image into another computed with the template matching method. By fitting a 2D elliptical Gaussian function, similarity values are created continuously and not in discrete (pixel) positions. In such a way, sub-pixel accuracy results, instead of pixel accuracy were computed.

To test the performance of the suggested sub-pixel accuracy method, the mutli-registration method was applied once with high resolution imagery and once with low resolution imagery but in this case also the sub-pixel accuracy method was employed. By doing so, it was observed that the accuracy of a large number of estimations computed with low resolution imagery was enhanced, but did not reach accuracy higher that then the results computed with high resolution imagery. For some cases though, the accuracy of the Gaussian fitting results was better than that of the results retrieved with the multi-registration of high resolution images. Therefore, it can be conclude that the method shows potential, but needs improvement.

It is believed that the sub-pixel accuracy method results that have bad quality are due to the following reasons:

1. According to the input point cloud tiles, the distribution of the similarity values in the score maps varies. Therefore, the selection of a single distribution is not an ideal solution.
2. The fact that an elliptical approximation is used is because the distribution of the values in the two main direction (such as  $\sigma_x$  and  $\sigma_y$ ) might be different. However, the spread ( $\sigma$ ) of the values in one direction is not the same in the both sides of the peak. For example, the  $(+\sigma_x)$  is different than the  $(-\sigma_x)$ . This is not taken into account in the elliptical Gaussian approximation.
3. The evaluation criteria are based on the (one) highest similarity value detected in the score maps. It is evaluated whether the difference between the

two highest peaks is large enough. However, it is possible that the difference between the two peaks is significant, but if the (Gaussian) surface fitting was applied to all the candidate peaks, it could be seen that the highest peak is found in another pixel than the one with the highest value (Figure 7.1).



**Figure 7.1:** The left part of the figure shows with blue bars the similarity scores of a score map. The black line illustrates an interpolation with a Gaussian fitting on the discrete pixel values. The right part of the figure shows again similarity scores of a score map. Although the peak value of the figure in left is higher than the peak value of the figure in the right, as soon as the fitting is performed, it is observed that with sub-pixel accuracy the highest similarity value is found at the right part.

By providing answers to the sub-questions, the main research question can be answered:

*To what extent is it possible to automatically, reliably, precisely and efficiently align mobile laser scanning data relatively, using an image-based technique?*

It is possible to use an image-based technique to align mobile laser scanning data relatively. The quality of the results, or in other words the reliability and the precision of the results given from the developed method, depend highly on the input data.

By testing different cases with high image resolution it is concluded that the suggested image-based registration approach is more likely to be **successful**, or in other words the results are more likely to have high quality, when:

- When there is some overlap between two point clouds AND,
- corresponding objects between the point clouds are distinct in pairs of 2D projections.

When those requirements are met, there is a high possibility that the developed method will register successfully point clouds with large translation errors (Figure 6.4), or even with some changes in one point cloud scene (Figure 6.1).

By testing different cases it is concluded that the suggested image-based registration approach is more likely to be **unsuccessful**, or in other words the results are more likely to have poor quality, when:

- There are parallel bridges and roads in a point cloud pair.
- The point cloud tiles share very small overlap.

The big advantage of the method is that even if the results have high or low quality (based on the reasons explained above), it provides the user with quality measures. Therefore the user, can specify thresholds on the quality measures, and subsequently employ the registration results only of the cases that have high quality. These would be beneficial for the **global registration** step, which follows the relative registration.

Moreover, by using the proposed image-based technique problems faced by the Iterative Closest Point, the commonly used method to register point clouds locally, are solved. Specifically, with the suggested method it is not needed to develop a method for finding the corresponding points in two point clouds which is an expensive process, especially if there are millions of points in the tile, which is the case with Mobile Laser Scanning data. With the proposed method it was also shown that there is no need for having initial good matching between two point clouds. This was shown with the cases where there was large offset between the point clouds and the method successfully matched them. Furthermore, ICP requires large to full overlap between the point clouds to produce matching results of high quality. This is needed because the ICP must construct correspondences between the point clouds, and if corresponding points do not exist, the algorithm is confused. As shown (for example in Figure 6.1), this is not necessary with the developed method. However, actual comparison of the proposed method with ICP was not conducted, but it is something considered for future work.

Finally, although image based registration methods are restricted to the pixel's accuracy, which is not the case with the 3D registration methods, a method for sub-pixel accuracy result was developed. Currently the method is not completely successful, but shows potential.

## 7.2 FUTURE WORK

### 7.2.1 Recommendations to improve the method

#### OUTLIER REJECTION

- Experiments should be executed to find out the ideal LOF threshold, as the number of rejected points is dependent on that. Important factor for the selection is how the quality of the registration results is affected.
- The outliers detection method could be improved by fitting planes to the points and rejecting those on which planes cannot be fitted. In doing so, points that represent objects that can possibly hinder the registration, such as trees, are eliminated.

#### NORMAL VECTORS COMPUTATION

- The current normal computation method is extremely slow. A way to change this would be to reduce the number of points as the computation of the normals is computed for every point. To do this, two things can be tried. First, the threshold for the outlier rejection method could be increased so as to remove larger number of points. The second way would be to use a down-sampling technique in order to reduce the amount of points as they usually have high density. However, for both techniques, not more points than needed for the registration should be removed.

If the time factor is of high importance, then the normal images could be completely discarded and use only the image types that produce good quality results. These are the density, intensity gradient and depth gradient images.

#### LIDAR DISTANCE COMPENSATION

- There must be compensation for the vehicle's distance to the scanned points. The values of the density and intensity currently used for the created imagery are relative. They are relative to the distance between the car and the scanned surfaces. For example, surfaces closer to the recording vehicle will be represented by a very high amount of points. In contrast, surfaces further

away will be scanned with significantly less number of points. There must be a balance between the two cases. This could be solved by sub-sampling in such a way that the resulting point cloud has a more uniform distribution. However, in that case it is necessary to make sure that points far away from the recording vehicle do not become more important than they are, because the further the points from the scanner, the lower their positional accuracy.

#### STATISTICAL MEASURES FOR THE PROVISION OF A SINGLE PIXEL VALUE

- Since imagery is created from point cloud data, many attributes of the points are compressed into single values. Currently the developed method uses measures of which the visual preliminary results were satisfactory, but detailed analysis of what is optimum must be conducted.

#### ADDITIONAL PIXEL VALUES TO CONSTRUCT REFERENCE IMAGES

- The created image pairs have equal size. To perform the template matching method it is necessary to generate search space, therefore pixels are added around one of the two images. Adding zero values to construct the reference image was an easy decision in order to achieve the goal. Firstly, thorough analysis must be performed to reveal if this decision deteriorates the quality of the registration results and if yes, then a more sophisticated approach must be developed.

#### EVALUATION CRITERIA

- The strength and uniqueness of the highest similarity value in the score maps is based on normalized pixel values. The current way with which the evaluation criteria of the score maps are developed stores the score map images and then evaluates them. The maximum value that can be stored with the OpenCV library which was used to write images is 65535. However, the similarity values can be higher than these value. The normalization of the values in the score maps up to 65535 may lead to false conclusions when performing the evaluation criteria. An example is when the two highest similarity values of a score map are compared to determine whether their difference is significant (see page 68), but their difference has been shrunk due to the normalization. Currently, many different thresholds have been test to try to overcome this problem. A way to solve this would be to apply the evaluation criteria before storing the score map images.
- Many score maps retrieved from density images were discarded because the X in XZ projections and the Y in YZ projections was not computed reliably, in comparison with the computation of the Z parameter. This is illustrated in the score maps as a single bright line in the Z direction. The currently implemented evaluation criteria, assess for the existence of a peak pixel, and therefore the aforementioned score maps are rejected, although they produce successful Z translations. A way to overcome this would be to alternate the evaluation criteria, detect areas of pixels with high value and compute their variance in the two directions of the projection. If the variance is small then the parameter is accepted. If the variance if large, then the parameter is rejected.

#### CALCULATION OF ALGORITHM'S EXECUTION TIME

- When calculating computation times it is better to split the process in several individual steps because then it is clear which steps require long execution time. In such a way, it is evident which steps must be improved.

#### EXTERNAL RELIABILITY

- Currently only one cause of possible undetected errors is taken into account to provide external reliability weights for each estimation. This is the number of points that exist in each point cloud. Other causes of possible undetected errors must be considered. For example the differences between overlapping point clouds due to moving objects and the existence of parallel objects with different elevations (such as the case with the bridge and the road, Figure 6.6).

#### 7.2.2 Recommendations to extend the project

##### ICP COMPARISON

- The proposed method is often verbally compared with the Iterative Closest Point algorithm but a comparison is not performed yet. Therefore it is something taken into account for future work.

##### WEIGHTS FOR THE ACCURACY OF A POINT CLOUD'S ABSOLUTE POSITION

- What has been developed in this thesis is a method that allows to compute the relative positioning errors between the point clouds. This method does not provide any information about the absolute accuracy of the matched point cloud pairs. As explained in Section 2.1.2, there is not information about the positional accuracy of the overlapping point clouds. However, it is necessary to find information about the positional accuracy of the overlapping point clouds, in order to know which point cloud's absolute position can be trusted more. In such a way it will be possible to distribute the translation error between the overlapping point clouds or provide this information as reliability weights for a Global registration approach (see next paragraph).

The GNSS observations recorded when scanning a point cloud cannot be trusted, because even if for example a satellite's observation was received by the scanner, it might be the case that the signal of this satellite was blocked or reflected on buildings, trees etc. Otherwise the GNSS observations could have been used to provide positional accuracy weights to each point cloud, based on the amount of received satellite signals. What it could be done to provide weights about the trustworthiness of a point cloud's absolute position is to construct GNSS visibility predictions (based e.g. on a 3D model, or panoramic images) and compare them with GNSS visibility observations in order to find out if the observations are representative of the truth. Then a scoring scheme can be used to assign similarity weights between predicted and observed satellite signals. If for example a signal is predicted to be invisible, but according to the GNSS observations at the recording time the signal was visible, then this is an indication that the signal was diffracted, however it was considered for the computation of a point cloud's position. Consequently, the specific point cloud should get a low accuracy reliability weight.

##### GLOBAL REGISTRATION

- A [global registration](#) follows the relative registration approach. After the relative transformations have been computed, they must be used in order to align all the point clouds scanned from different epochs together and construct a complete 3D model of an area. To do this, a Pose Graph Optimization (PGO) technique could be used. A PGO is the formulation of a graph whose nodes correspond to the poses of the vehicle at different points in time and whose edges represent constraints between the poses. The positions of the moving platform will be given from the [GNSS](#) and also from the [IMU](#) observations, and thus the distances between the poses could be used

as constraints. Also, the individual local registration solutions could be used as positioning constraints within the pose graph. Then, trustworthiness weights (as explained in the previous paragraph) should be attached to each of the three constraints; the GNSS, the IMU and the local registration solutions. Once such a graph is constructed, the global solution can be computed by finding the spatial configuration of the nodes that is mostly consistent with the measurements modeled by the edges (constraints) [Grisetti et al., 2010]. In other words, a pose graph attempts to pull the positions of the scans by distributing the positioning error based on the input information so that an optimal graph of the scans' positions is obtained. As a result, a complete 3D point cloud model could be retrieved.



## BIBLIOGRAPHY

- Anthony, S. M. and Granick, S. (2009). Image Analysis with Rapid and Accurate Two-Dimensional Gaussian Fitting. *Langmuir*, 25(14):8152–8160.
- Baarda, W. (1968). *A testing procedure for use in geodetic networks. New Series, vol 2, no 5. Netherlands Geodetic Commission, Publications on Geodesy New Series. Delft, The Netherlands. Kanaalweg 4, Rijkscommissie voor Geodesie.*
- Barshan, B. and Durrant-Whyte, H. F. (1995). Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342.
- Bellekens, B., Spruyt, V., Berkvens, R., and Weyn, M. (2014). A Survey of Rigid 3d Pointcloud Registration Algorithms.
- Bentley, J. L. (1975). Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM*, 18(9):509–517.
- Beraldin, J.-A. (2004). Integration of Laser Scanning and Close-Range Photogrammetry - The Last Decade and Beyond. *Proceedings of the XXth ISPRS Congress.*
- Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.
- Blomley, R., Weinmann, M., Leitloff, J., and Jutzi, B. (2014). Shape distribution features for point cloud analysis - a geometric histogram approach on multiple scales. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:9–16.
- Bradski, G. and Kaehler, A. (2008). *Learning OpenCV.*
- Breunig, M. M., Kriegel, H. P., Ng, R. T., and Sander, J. (2000). LOF: Identifying Density-based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00*, pages 93–104, New York, NY, USA. ACM.
- Brown, G. and Butler, H. (2012). Laspy: Documentation — laspy 1.2.5 documentation.
- Byun, S., Jung, K., Im, S., and Chang, M. (2017). Registration of 3d scan data using image reprojection. *International Journal of Precision Engineering and Manufacturing*, 18(9):1221–1229.
- Cartwright, W., Gartner, G., Meng, L., and Peterson, M.P (2009). *Lecture Notes in Geoinformation and Cartography, Extraction of Features from Mobile Laser Scanning Data for Future Driver Assistance Systems.* Springer.
- Chan, T. O. and Lichti, D. D. (2015). Automatic In Situ Calibration of a Spinning Beam LiDAR System in Static and Kinematic Modes. *Remote Sensing*, 7(8):10480–10500.
- Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155.
- Chetverikov, D., Svirko, D., Stepanov, D., and Krsek, P. (2002). The Trimmed Iterative Closest Point algorithm. In *Object recognition supported by user interaction for service robots*, volume 3, pages 545–548.

- Chmelar, P., Beran, L., and Rejcek, L. (2016). The Depth Map Construction from a 3d Point Cloud. *MATEC Web of Conferences*, 75:03005.
- Christodoulou, A. (2017). An image registration technique for 3d point cloud data alignment. Technical report.
- Cizek, P. and Visek, J. (2000). Least Trimmed Squares. In *XploRe® — Application Guide*, pages 49–63. Springer, Berlin, Heidelberg.
- Craven, P. V. (2016). Chapter 5: Introduction to Graphics. 5.1 Computer Coordinate Systems. In *Program Arcade Games With Python And Pygame*.
- Ding, L., Goshtasby, A., and Satter, M. (2001). Volume image registration by template matching. *Image and Vision Computing*, 19(12):821–832.
- Flores-Mangas, F. (2014). The image gradient. Course CSC320w: Introduction to Visual Computing. University of Toronto.
- Gaidhane, V. H., Hote, Y. V., and Singh, V. (2014). A computationally efficient approach for template matching-based image registration. *Sadhana*, 39(2):317–331.
- Gnedenko, B., Pavlov, I. V., and Ushakov, I. A. (1999). *Statistical Reliability Engineering*. John Wiley & Sons.
- Godin, G., Rioux, M., and Baribeau, R. (1994). Three-dimensional registration using range and intensity information. In *Proc. SPIE 2350, Videometrics III*.
- Grisetti, G., Kummerle, R., Stachniss, C., and Burgard, W. (2010). A Tutorial on Graph-Based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43.
- Grous, A. (2013). *Analysis of Reliability and Quality Control: Fracture Mechanics 1*. Wiley, Hoboken, UNITED STATES.
- Haala, N., Peter, M., Kremer, J., and Hunter, G. (2008). Mobile LIDAR Mapping for 3d Point Cloud Collection in Urban Areas - A Performance Test. In: *Proceedings of XXI ISPRS Congress, Beijing, China, July 3-11, 2008*, 37.
- Hampel, F. R. (1971). A General Qualitative Definition of Robustness. *The Annals of Mathematical Statistics*, 42(6):1887–1896.
- Huang, Q.-X., Adams, B., Wicke, M., and Guibas, L. J. (2008). Non-rigid Registration Under Isometric Deformations. In *Proceedings of the Symposium on Geometry Processing, SGP '08*, pages 1449–1457, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- Huang, X., Zhang, J., Fan, L., Wu, Q., and Yuan, C. (2017). A Systematic Approach for Cross-source Point Cloud Registration by Preserving Macro and Micro Structures. *IEEE Transactions on Image Processing*, 26(7):3261–3276. arXiv: 1608.05143.
- Huber, D. F. and Hebert, M. (2003). Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(7):637 – 650.
- Höfle, B. and Pfeifer, N. (2007). Correction of laser scanning intensity data: Data and model-driven approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(6):415–433.
- Isenburg, M. (2015). Five Myths about LAS, LAZ, and “Optimized LAS”.
- Jacobs, D. (2005). Image Gradients.
- Joosten, P. (2017). Personal communication.

- Kaartinen, H., Hyypä, J., Kukko, A., Jaakkola, A., and Hyypä, H. (2012). Benchmarking the Performance of Mobile Laser Scanning Systems Using a Permanent Test Field. *Sensors*, 12(9):12814–12835.
- Kang, Z., Li, J., Zhang, L., Zhao, Q., and Zlatanova, S. (2009). Automatic Registration of Terrestrial Laser Scanning Point Clouds using Panoramic Reflectance Images. *Sensors*, 9(4):2621–2646.
- King, B. J., Malisiewicz, T., Stewart, C. V., and Radke, R. J. (2005). Registration of multiple range scans as a location recognition problem: hypothesis generation, refinement and verification. In *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pages 180–187.
- Kozubowski, T. J., Podgórski, K., and Rychlik, I. (2013). Multivariate generalized Laplace distribution and related random fields. *Journal of Multivariate Analysis*, 113:59–72.
- Krishnan, S., Lee, P. Y., Moore, J. B., and Venkatasubramanian, S. (2007). Optimisation-on-a-manifold for global registration of multiple 3d point sets. *International Journal of Intelligent Systems Technologies and Applications*, 3(3-4):319–340.
- Kukko, A., Kaartinen, H., Hyypä, J., and Chen, Y. (2012). Multiplatform Mobile Laser Scanning: Usability and Performance. *Sensors*, 12(9):11712–11733.
- Langerwisch, M. and Wagner, B. (2010). *Registration of Indoor 3D Range Images using Virtual 2D Scans*.
- Lavrenko, V. (2015). kNN.15 K-d tree algorithm.
- Lemmens, M. (1992). Image Matching Strategies for the Co-registration of CAESAR Sea Images: An Hierarchical Aiea-Based Approach. The Netherlands Remote Sensing Board. volume Part II.
- Lemmens, M. (2011a). Global Navigation Satellite Systems and Inertial Navigation. In *Geo-information, Geotechnologies and the Environment*, pages 55–83. Springer, Dordrecht.
- Lemmens, M. (2011b). Terrestrial Laser Scanning. In *Geo-information, Geotechnologies and the Environment*, pages 101–121. Springer, Dordrecht.
- Lemmens, M. J. (1988). A survey on stereo matching techniques.
- Levi, R. W. and Judd, T. (1996). Dead reckoning navigational system using accelerometer to measure foot impacts.
- Liang, Y. B., Qiu, Y., and Cui, T. J. (2017). Semiautomatic Registration of Terrestrial Laser Scanning Data Using Perspective Intensity Images. *IEEE Geoscience and Remote Sensing Letters*, 14(1):28–32.
- Lin, C.-C., Tai, Y.-C., Lee, J.-J., and Chen, Y.-S. (2017). A novel point cloud registration using 2d image features. *EURASIP Journal on Advances in Signal Processing*, 2017(1):5.
- Magnusson, M., Lilienthal, A., and Duckett, T. (2007). Scan registration for autonomous mining vehicles using 3d-NDT. *Journal of Field Robotics*, 24(10):803–827.
- Matkan, A. A., Hajeb, M., Mirbagheri, B., Sadeghian, S., and Ahmadi, M. (2014). Spatial Analysis for Outlier Removal from LIDAR Data. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:187–190.

- Mendenhall, S. (2014). Mobile laser scanning. *Civil and Structural Engineer magazine*.
- Naidu, D. K. and Fisher, R. B. (1991). A Comparative Analysis of Algorithms for Determining the Peak Position of a Stripe to Sub-pixel Accuracy. In *BMVC91*, pages 217–225. Springer, London.
- NCTech (2018). The LASiris VR: Enjoy Bleeding Edge 3d VR 360 Imaging.
- Neubeck, K. (2004). *Practical Reliability Analysis*. Prentice Hall.
- Norollah, M., Pourghassem, H., and Mahdavi-Nasab, H. (2012). Image Registration Using Template Matching and Similarity Measures for Dental Radiograph. In *Fourth International Conference on Computational Intelligence and Communication Networks*, pages 331–335.
- Nüchter, A., Lingemann, K., Hertzberg, J., and Surmann, H. (2005). Heuristic-Based Laser Scan Matching for Outdoor 6d SLAM. In *KI 2005: Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 304–319. Springer, Berlin, Heidelberg.
- Padia, C. and Pears, N. (2017). A Review and Characterization of ICP-based Symmetry Plane Localisation in 3d Face Data.
- Pang, G. (2011). *Outlier detection and data filtering in LiDAR data with multiple attributes*. Thesis, The Hong Kong Polytechnic University.
- Pfeifer, N., Dorninger, P., Haring, A., and Fan, H. (2007). Investigating terrestrial laser scanning intensity data: quality and functional relations. page 11.
- Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148.
- Puente, I., González-Jorge, H., Arias, P., and Armesto, J. (2011). Land-Based Mobile Laser Scanning Systems: a Review. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3812:163–168.
- Pulli, K. (1999). Multiview registration for large data sets. In *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062)*, pages 160–168.
- Rousseeuw, P. (1985). Multivariate Estimation With High Breakdown Point. *Mathematical Statistics and Applications Vol. B*, pages 283–297.
- Rousseeuw, P. J. (1984). Least Median of Squares Regression. *Journal of the American Statistical Association*, 79(388):871–880.
- Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF - IEEE Conference Publication. Barcelona, Spain. IEEE.
- Sanchez, J., Denis, F., Checchin, P., Dupont, F., and Trassoudaine, L. (2017). Global Registration of 3d LiDAR Point Clouds Based on Scene Features: Application to Structured Environments. *Remote Sensing*, 9(10):1014.
- Sarvaiya, J. N., Patnaik, S., and Bombaywala, S. (2009). Image Registration by Template Matching Using Normalized Cross-Correlation. In *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, pages 819–822.
- Scaramuzza, D., Harati, A., and Siegwart, R. (2007). Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4164–4169.

- Segal, A. V., Haehnel, D., and Thrun, S. (2009). Generalized-ICP. In *Proceedings of Robotics: Science and Systems.*, Seattle.
- Sharp, G. C., Lee, S. W., and Wehe, D. K. (2002). ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):90–102.
- Shetty, A. (2017). *GPS-LiDAR sensor fusion aided by 3D city models for UAVs*. PhD thesis, University of Illinois at Urbana-Champaign.
- Smith, L. (2002). A tutorial on Principal Components Analysis.
- Sotoodeh, S. (2006). Outlier detection in laser scanner point clouds. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI-5*, pages 297–302.
- Sweco Nederland B.V. (2016). MOVE3 Software User Manual, Version 4.4. <https://move3software.com/downloads/> (18 March 2016).
- Teunissen, P. (2000). *Adjustment theory*. Delft University of Technology, Department of Mathematical Geodesy and Positioning.
- Teunissen, P. J. G., Simons, D., and Tiberius, C. (2006). *Probability and Observation Theory*. Faculty of Aerospace Engineering. Technical University of Delft.
- Theiler, P. W., Wegner, J. D., and Schindler, K. (2015). Globally consistent registration of terrestrial laser scans via graph optimization. *ISPRS Journal of Photogrammetry and Remote Sensing*, 109:126–138.
- Trucco, E., Fusiello, A., and Roberto, V. (1999). Robust motion and correspondence of noisy 3-D point sets with missing data. *Pattern Recognition Letters*, 20(9):889–898.
- Van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., Tijssen, T., Kodde, M., and Gonçalves, R. (2014). Massive point cloud data management: design, implementation and execution of a point cloud benchmark. *Computers & Graphics*, 49:92–125.
- van Someren, B. (2016). Comparing Features for Point Cloud Registration with Varying Densities. Technical report.
- Velodyne LiDAR, Inc. (2010). High definition LiDAR HDL-32e.
- Velodyne LiDAR, Inc. (2012). User’s manual and programme guide. HDL-32e High Definition LiDAR™ Sensor.
- Wang C., Cho Y. K., and Park J. W. (2014). Performance Tests for Automatic 3d Geometric Data Registration Technique for Progressive As-Built Construction Site Modeling. *Computing in Civil and Building Engineering (2014)*.
- Zhang, G., Lei, M., and Liu, X. (2009). Novel template matching method with sub-pixel accuracy based on correlation and Fourier-Mellin transform. *Optical Engineering*, 48(5):057001.



# A

## REFLECTION

This appendix describes the value of this graduation thesis in the larger social and scientific framework.

The main characteristic of the methodical line of approach of the Geomatics Master at TU Delft is that the student has to dive into every project to deliver solutions that are based on detailed research so that every step taken can be justified. Similarly the method chosen to deliver this project was based on thorough research. What I learned from that is that even if there are easy solutions to perform certain tasks, if a student knows why is it beneficial to choose a specific solution then there is higher probability to obtain results of good quality. This way of thinking will definitely help me to accomplish future projects.

The following paragraph explains the relationship between the conducted research and application of the field geomatics. The research performed in this project considers one step of the many that must be followed to deliver complete and geo-referenced point clouds. Many people are impressed when looking at point clouds, but a few people are aware of the usefulness of such data. In this project I mention application which can be developed when having registered point clouds. In my opinion, the most important application that can be achieved with registered point clouds is sitting at your desk and being able to measure coordinates, distances, volumes, detecting changes between destroyed areas and many others just from the comfort of your desk. Work which would need days and many people to be done, it is possible to be achieved by having registered georeferenced 3D point clouds. For the scientific world of Geomatics the developed method can be considered a big contribution as many people are currently using the ICP algorithm which has quality and efficiency limitations.

## COLOPHON

This document was typeset using  $\text{\LaTeX}$ . The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from André Miede.

